

Notes on Real-Time Vehicle Simulation

**Roderic Deyo
David Ingebretsen**

**New Business Development
Evans & Sutherland
Salt Lake City, Utah 84108**

©1989 Roderic C. Deyo

May 5, 1989

Contents

1	The Driving Simulator System	7
1.1	Introduction	7
1.2	Historical Perspective	8
1.3	Today's Advanced Simulator	9
1.4	Relating Graphics and Dynamics Under Real-Time Constraints	12
1.5	CIG Demands of High-Performance Driving Simulation	14
2	Dynamics	17
2.1	Dynamics for Computer Simulation	17
2.2	Review of Newtonian Dynamics	18
2.2.1	Newton's Laws of Motion	18
2.2.1.1	Coordinate Systems	19
2.2.1.2	Newton's Second Law	19
2.2.1.3	Inertial Reference Frames	21
2.2.1.4	Newton's First Law	21
2.2.1.5	Newton's Third Law	22
2.2.1.6	Angular Momentum	22
2.2.1.7	Rate of Change of Angular Momentum	24
2.2.2	Multiple Particles and Constraints	24
2.2.2.1	Holonomic Constraints	24
2.2.2.2	Multiple Constraints	27
2.2.2.3	Constraint Jacobian	28
2.2.3	Principle of Virtual Work	29
2.2.3.1	Statement of Virtual Work Principle .	30
2.2.3.2	Relation with other Variational Methods	30
2.2.3.3	Derivation of Virtual Work Principle	31

2.2.3.4	D'Alembert's Principle	33
2.2.3.5	Kinematically Admissible Virtual Displacements	33
2.2.3.6	An Essential Difficulty	35
2.2.4	Generalized Coordinates	36
2.2.4.1	Change of Variables to Generalized Coordinates	38
2.2.4.2	Chain Rule	39
2.2.4.3	Kinematically Admissible Generalized Virtual Displacements	41
2.2.4.4	D'Alembert's Principle in Generalized Coordinates	41
2.2.5	Lagrange Multipliers	43
2.2.5.1	Lagrange Multiplier Theorem	43
2.2.5.2	Application of the Theorem	44
2.2.5.3	Matrix Formulation of Constrained Equations of Motion	45
2.3	Rigid Body Kinematics	47
2.3.1	The Rigid Body	47
2.3.1.1	Definition of a Rigid Body	47
2.3.1.2	Center of Mass	48
2.3.1.3	Rigid Body Orientation	49
2.3.2	The Description of Rotations	51
2.3.2.1	Rotations as Orthogonal Matrices	52
2.3.2.2	Euler Angles	52
2.3.2.3	Finite Rotation Formula	54
2.3.2.4	Euler Parameters	55
2.3.2.5	Infinitesimal Rotations	57
2.3.3	Rate of Change of a Vector	57
2.3.3.1	Infinitesimal Rotational Relations	58
2.3.3.2	Vector Rate of Change Equation	58
2.3.3.3	Velocity and Acceleration	58
2.3.4	Euler's and Chasles' Theorems	59
2.4	Dynamics of Rigid Bodies	61
2.4.1	Rigid Body Motion	61
2.4.2	Angular Momentum	62
2.4.3	Inertia Tensor	63

<u>Evans & Sutherland Proprietary</u>	3
2.4.3.1 Newton-Euler Equations	64
2.4.3.2 Virtual Work Equation for a Rigid Body	70
2.4.3.3 Recipe for Multi-body Equations of Motion	72
3 Real-Time Dynamics Algorithms	75
3.1 Previous Dynamics Modeling Techniques	75
3.2 Derivation of Recursive Dynamics Equations for Vehicle Model	76
3.2.1 Rigid Body Kinematics and Euler Parameters .	76
3.2.2 Kinematic Relations - The Vehicle Graph . . .	77
3.2.3 Virtual Work Equations	82
3.2.4 Recovery of Accelerations from the Recursive Equations	84
3.3 Vehicle Dynamics Model	87
3.3.1 Real-Time Dynamics for Vehicle Simulation . .	87
3.3.2 Vehicle Description	89
3.3.3 Real-time Algorithm	91
3.3.3.1 Decomposition in terms of a Vehicle Graph	91
3.3.3.2 Vehicle Coordinates	92
3.3.3.3 Holonomic Vehicle Constraints	93
3.3.3.4 Recovery of Outboard Positions, Velocities, Accelerations	94
3.3.3.5 Recovery of Coordinate Accelerations	94
3.3.3.6 External Forces	95
3.3.3.7 Integration of Accelerations	96
3.3.3.8 Constraint Stabilization	97
3.3.4 Organization of Parallel Recursive Algorithm .	98
3.3.4.1 Algorithm Flow	98
3.3.4.2 Computer Architectures and Software	100
3.3.4.3 Algorithm Timing	103
3.3.4.4 Remarks	103
4 Physiology for Simulation	105
4.1 Effective Simulation	105
4.1.1 Linear Systems Theory	105

4.1.2	Human Perception Systems	107
4.1.2.1	The Eyes	107
4.1.2.2	The Vestibular System	108
4.1.2.3	Perception Thresholds	112
4.1.2.4	Proprioceptive and Tactile Sensors . .	114
4.1.3	Simulator Sickness	115
4.1.4	Designing an Effective Simulator Motion System	117
4.1.4.1	Washout Filters	118
4.1.4.2	Summary	125
4.1.5	Control Force Loading	125
4.1.5.1	Vehicle Commands and Controls . . .	126
4.1.6	The Total System	127
4.1.7	The Future	128
5	References	129
5.1	References on Dynamics	129
5.2	References on Perception	131

Forward

Driving simulation requires the construction of a total synthetic real-time environment. An effective simulator must provide proper motion, visual, sound, and tactile cueing for the driver to respond in a natural manner. A simulated driving environment is an excellent example of a complex "real-time microworld", where a person must interact completely and naturally with an artificial, but compelling, computer controlled world.

Driving simulation is an interesting case study because it entails synthesizing a familiar, yet highly non-trivial, environment. It must provide accurate vehicle dynamics predicting real vehicle behavior; graphics which provide sufficient optical flow density for proper control and speed-distance judgments by the subject; optimal acceleration cueing; sound and control force feedback cueing mechanisms to enhance the sense of realism for the subject. The goal of human interaction is to provide a designer with information about performance, stability, ride quality, and safety margins to suit human tolerances in control, task loading, and comfort.

Our purpose in writing these notes is to highlight the essential system issues for the design and implementation of real-time, animated environments for vehicle (and, in particular, driving) simulation.

To meet these goals, these notes introduce ideas about

- Total simulator design.
- Real-time vehicle dynamics.
- Host computer and parallel processing issues.
- Physiological models of perception for cueing optimization.
- Washout algorithm design for matching motion system capabilities.
- Control force loading.

To know who to blame, the first author (R.D.) was primarily responsible for the first three chapters on dynamics and the real-time

vehicle algorithm, while the second author (D.I.) was primarily responsible for chapter four on physiological considerations.

We want to thank the members of the Driving Simulation Project at Evans & Sutherland for their efforts towards real-time vehicle simulation. Pete Doenges and John Briggs have provided the vision, conceptual foundations for advanced driving simulation, and part of the text from their Siggraph paper ([22]). We also want to thank E. Haug, D-S. Bae, R-S. Hwang, D. Andersen, and W-S. Lee. Any errors in these notes are, of course, our own contribution.

Chapter 1

The Driving Simulator System

1.1 Introduction

In the last fifteen or so years, man-in-the-loop simulation using computer-generated, real-time graphics has assumed a major role not only in pilot training, but also in engineering applications for land, sea, air, and space vehicles. Man-in-the-loop simulation uses a person in the control loop of a simulation to provide feedback to the system operations. The operator experiences a synthetic environment, manipulates his controls in response to the situation, and experiences the results of his actions. In a real-time simulator, the dynamics model is continually evaluated to give feedback to the operator. On the basis of this data, the simulator provides realistic environmental cues to allow him to control the simulated vehicle in the same way that he would in the real world.

The goal of human interaction with a design is to tune raw performance, stability, ride quality, and safety margins, to suit human tolerances in control, task loading, and comfort. Through the use of simulation, training or design testing can often be achieved at lower cost, experiments can be conducted under controlled, repeatable circumstances, and — important for engineering — vehicle designs can be evaluated while minimizing the number of prototypes.

An important engineering use of simulation with a person in the

loop is to determine vehicle response and to study the man-machine interface before the start of a vehicle test program. For example, a new aircraft can be modeled and test flown even before it leaves the drawing board. The pilots can give valuable feedback to the designers about the plane's characteristics at an early enough stage to permit changes in the airframe. Integration of the avionics systems can be achieved at an earlier time. Workload measurements can be made, and different alternatives can be tested. Computer graphics represents a large portion of the cost of man-in-the-loop simulation. Hence, as the capabilities of computer image generation have improved and the relative cost of simulation has dropped, there are better economic justifications for its use. In some applications, such as the space shuttle simulator, there has never been a good alternative to simulator training. In other areas, however, such as vehicle engineering, cost of man-in-the-loop simulation testing has always been an important consideration, and, as a result, has only recently been low enough to justify simulation on a large scale.

Computer graphics is the critical element in creating a realistic operator environment in a simulator, but the graphics must work in concert with the rest of the system to be believable. In particular, the vehicle motion and the scenes generated must seem realistic in order not to seem like an arcade game. To show how real-time computer graphics and dynamics must work together in vehicle simulation, we have selected a specific application to discuss — driving simulation. We will give a brief history of the use of synthetic imagery in simulation, identify the components of a modern simulator with emphasis on the role of the visual system, present a new approach to computing vehicle dynamics that promises significantly improved vehicle models for realistic motion, and conclude with a view of the future.

1.2 Historical Perspective

Department of Defense initiatives through the military and its contractors have done much to advance the state-of-the-art in simulation and computer graphics. For example, seminal work in computer graphics conducted at the University of Utah under the leadership

of Dr. David Evans two decades ago was largely sponsored by the Department of Defense's Advanced Research Projects Agency.

The U. S. Air Force supported in part the first use of CRTs in driving simulation at MIT in 1962 [63]. This work by T.B. Sheridan, H.M. Paynter, and S.A. Coons was the earliest use of an artificial display of a vehicle path in dynamic perspective. A refined implementation was developed by W.W. Wierwille, G. A. Gagne, and J. Knight at Cornell Aeronautical Laboratory in 1966 [69].

Use of these same techniques occurred at General Motors in the late 1960's. In the early 1970's, both Volkswagen [59] and the Swedish Road and Traffic Institute (VTI) [52] built driving simulators with visual and motion subsystems. A number of simulators were built over the next decade. For example, simulators at the Virginia Polytechnic Institute and State University [17] and Deere and Company [26] were developed primarily to support human factors research. Another system, complete with motion platform, was constructed at Universitaet der Bundeswehr Muenchen for research into guiding land vehicles with computer vision — a sort of robot-in-the-loop simulation [23]. The graphics associated with these systems were limited. Line segments representing the road were displayed on a single CRT or projected on a flat screen in front of the driver.

In 1984, over twenty years after the MIT project, Daimler-Benz built, in Berlin, the first driving simulator expressly as an aid to designing vehicles [27] [33]. This system marked the first use of a 180-degree-wide image on a dome in a driving simulator. For the first time a driver was able to use his peripheral vision in a natural way. The Daimler-Benz simulator was built to allow test drivers to make subjective judgments about the handling qualities of new automobile designs.

1.3 Today's Advanced Simulator

An advanced man-in-the-loop simulator vehicle (Figure 1.1) is made up of a few, expensive subsystems that combine to create a complete synthetic driving environment for the operator:

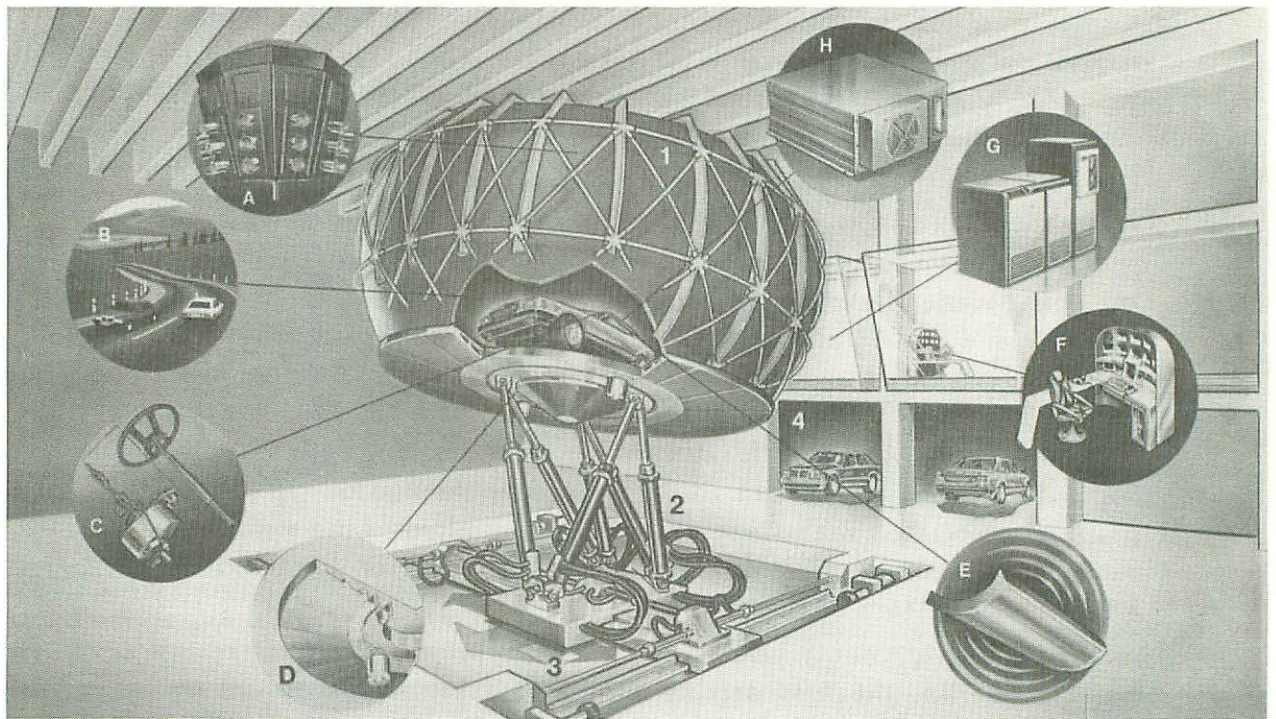


Figure 1.1: Modern advanced vehicle simulator system.

- The host computer (Figure 1g) controls all of the real-time operations of the system, including computation of the vehicle and motion platform dynamics. All calculations for the entire system are performed within a 10 ms time frame. An integrated data base contains visual models, geometric data on the simulated environment, data relating to the sounds encountered, and mechanical characteristics of the vehicle.
- The visual system calculates the correct image (Figure 1b) based on position information from the host computer. It removes hidden areas, computes the sun angle, smooth shades, textures, and anti-aliases the result. There is a high degree of parallelism in the foregoing operations, allowing about ten GIPS (Giga Instructions Per Second) operation in a large system.
- Multiple CRT projectors (Figure 1a) display an image on the dome. Special screen material is used for maximum brightness and contrast.
- The motion platform (Figure 1-2,3,d) is a configuration of hydraulic actuators that can move the operator's cab in any direction or angle by combining motions about the x, y, z, yaw, pitch, and roll axes. The control signals to the motion platform are filtered ("washed out") to transform the virtual real-world motion computed by the dynamics software into movements achievable by the system.
- The sound subsystem (Figure 1e) uses multiple speakers to generate appropriate sounds to simulate the engine, gear whine, wind noise, passing traffic, etc.
- Instrumentation (Figure 1h) provides data to the host computer to monitor the the entire system and ensure the driver's safety.
- Control force loaders (Figure 1c) are used to provide appropriate forces back to the operator through the steering wheel, brake pedal, and transmission lever to replicate the feedback forces found in a real car.

- The operations console (Figure 1f) allows a single person to oversee all of the simulation activities, allowing flexible control over vehicle tests.

1.4 Relating Graphics and Dynamics Under Real-Time Constraints

Driving simulation for automotive engineering must provide enough accuracy in simulating vehicle motion and in subsystems cueing (visual, sound, motion, control force feedback, and instruments) to stimulate realistic driver responses that accurately portray flaws or inadequacies in the "paper" design of the target vehicle. The fidelities of graphics and dynamics crucial to driver cueing are measured by different yardsticks than those employed in non-real-time modeling. For example, the graphics realism of ray-tracing or total correlation between dynamics response from instrumented cars and vehicle models needing overnight computer runs are not necessarily pertinent.

Fully interactive real-time simulation demands care in the relative realism of the environmental scene (scene specificity and density) and dynamics accuracy significant to the driver response (chassis accelerations and control forces). Also important is the rate at which new cues are delivered to the driver and the time lag between his control commands and the corresponding cue changes. What is the proper balance between real-time constraints and the cueing rates needed to support a driver's decision process? What is the allowable system time lag before false driver responses are induced?

The subtlety of human perception and response while interacting with a vehicle forces difficult compromises. The dynamics considerations bearing on driver behavior demand an almost open-ended fidelity escalation. A driver's sensitivities to visual and tactile cues tend to peak in the region of a few hertz while steering in response to sudden external stimuli and at tens of hertz for periodic and impulsive road features. Rates of cueing tend to cap the response bandwidth of simulator vehicle dynamics. Much higher frequencies can arise from mechanical resonances of suspension parts and body torsion/flex, or the dynamic response of an on-board digital control system. How will

the dynamics simulation deal with cueing-significant modal energy which is coupled by system non-linearities from these higher frequency ranges down into the ranges of driver sensitivities?

Simulation-aided engineering requires vehicle designers to trust simulation realism when no equivalent physical vehicle system has been prototyped or tested. Computer-aided design and analysis tools can often supply reliable subsystem data. Since the total system can behave differently from the sum of its parts, how can the designer trust a simulator whose response might compromise the otherwise adequate subsystem data?

The answers to all these tough questions can be summarized with a few rules of thumb:

- Provide flexibility in trading system resources to enhance temporarily the specific areas of experimental focus at the expense of others.
- Ensure that the minimum requirements of human perceptual cueing thresholds and of vehicle operations are met by simulator performance and functionality.
- Attempt to drive overall simulator response as close to the step-steering and steady state (in both phase and frequency) responses of the vehicle as practical.
- Provide constant interfaces between different forms of the same data distributed within the simulator so that changes afforded by technology advances can be easily incorporated into simulator improvements.
- Develop simulator software architectures that port easily into future computer systems.
- Provide as much reserve capability as economically possible.

1.5 CIG Demands of High-Performance Driving Simulation

The use of real-time, man-in-the-loop simulators for the evaluation of designs of high-performance land vehicles has greatly lagged flight simulation. Major advances in dynamics, washout, motion, and graphics have recently set the stage for another leap.

The current state of CIG (Computer Image Generation) evolution is closely approaching photographic realism with update rates, lag, and image quality that suffice for a broad range of operational and perceptual objectives. Recent advances in computing and mechanical dynamics techniques have set the stage to achieve vehicle behavior in a simulator rivaling the quality of CIG real-time graphics, and to afford dynamics quality capable of unmasking subtle CIG shortcomings rarely encountered before now.

CIG improvement is driven by operational maneuvering and perceptual requirements [24]. Driving involves very low eye heights where optical flow density must change very rapidly over the field of vision available in a car. The driver must be able to judge speed and proximity to obstacles very quickly by visualizing textural cues in and around the road as well as passing 3D features. The visual depiction of the road surface must be good enough for the driver to anticipate road hazards like bumps and ice. When appropriate, urban, country, and highway scenes must be possible with contending traffic and pedestrians to saturate the driver's task loading and reaction times.

The basic psychology of vision and kinematics of driver responses at the controls of the car also demand CIG fidelity. The effective resolution of the CIG image determines what the driver can detect and recognize. A driving simulator should ideally match the lag, from control input to visual result, at a few tens of milliseconds typical of responsive cars. The rate of image refresh and update must satisfy eye demands for visual tracking, smooth motion of scenery, flicker-free viewing and discrimination of temporal events that may alter driver response.

The current generation of top-end real-time CIG is represented by such products as the Evans & Sutherland ESIG-1000. This system



Figure 1.2: Advanced real-time display. Example of high level of detail in modern real-time image generation using textures.

offers critical improvements in scene performance needed to discriminate the nuances of vehicle behavior possible with the dynamics techniques covered in the following section. Synthetic and photographic texture along with high 3D feature densities now allow sound judgments to be made about speed and distance. Subpixel anti-aliasing allows road signs and road surfaces to be visualized reliably at varied vehicle speeds. Generalized occultation allows complex traffic movement over rolling terrain, around urban clutter, and among other traffic. Multiple eyepoints can support the virtual eyepoints of mirrors and the simultaneous observation of own-vehicle chassis and suspension behavior from external vantage points while driving the car through difficult maneuvers.

Figure 1.2 shows a typical driving landscape and situation achievable with ESIG-1000 for driving simulation. Figure 1.3 represents the kind of car-external viewing of critical moments in suspension deformation and chassis excursions. This real-time visual display can complement a designer's study of subtle car subsystem motion over specific road geometries.



Figure 1.3: Examples of real-time generation of an external view of the vehicle motion showing suspension deformation. Frames are 80 ms apart.

Chapter 2

Dynamics

2.1 Dynamics for Computer Simulation

This chapter of the the notes shows how to make complex multi-body vehicles in the microworld move correctly in real-time. It reviews the basic concepts of rigid body kinematics and dynamics, including the virtual work equations and Lagrange multipliers. The next chapter will apply these ideas in the form of Bae's recursive algorithm for linked systems of rigid bodies. This algorithm has been shown to be the fastest available for either serial or shared-memory parallel processors, and its description is the goal of the dynamics part of these notes.

The following topics are covered in the next two chapters dealing with real-time vehicle dynamics and its implementation for a driving simulator:

- Quick review of basic particle and rigid body dynamics.
 - Newton's Laws. Linear and angular momentum.
 - Virtual work principle. Constraints. Generalized coordinates.
 - Rigid body kinematics. Transformation matrices. Euler angles and parameters.
 - Newton-Euler equations for a rigid body. Example of symmetric top. Constrained virtual work equations for systems

of rigid bodies.

- Real-time dynamics algorithms.
 - Use of system graph to decompose linked rigid body vehicles. Example of car.
 - Description of Bae's recursive algorithm for real-time dynamics of linked, multi-body systems. Example of car.
 - Parallel processing issues. Alliant FX/80 example.
- Practicalities. It will discuss the following topics:
 - Real-time integration methods. Using an eigenvalue analysis.
 - Force models.

Newton started by considering the motion of planets and projectiles. Later investigators considered rigid bodies. Modern dynamics is mainly concerned with complex systems of linked, constrained rigid and flexible bodies, moving under the influence of external forces and moments. Constrained multi-body dynamics is an active area of current research. Its roots go well back to the late eighteenth and early nineteenth centuries. The theoretical foundations are relatively well understood, with many possible approaches. For complex systems, few analytic models could (or can) be solved. With the advent of computers, involved models can now be done easily in non-real-time. The current challenge is to solve these same models in real-time to allow man-in-the-loop interactions.

2.2 Review of Newtonian Dynamics

In this section we review the concepts from classical mechanics needed to understand the vehicle dynamics algorithms.

2.2.1 Newton's Laws of Motion

Newton's laws originated in considering planetary motion. An individual planet is a body moving in space subject to various known

gravitational forces from the sun, other planets, and any other bodies having mass. Newton's original concern with multi-body dynamics was to derive Kepler's laws of planetary motion from first principles, i.e. Newton's law of universal gravitation. One of his first theorems showed that a planet was, for purposes at hand, equivalent to a particle with the total planet mass.

Our job is more complex, since we must deal with bodies which do not act like point particles; our bodies will have rotational, as well as translational, inertia about their center-of-mass. In addition, these bodies are linked in complex kinematic configurations, where relative motion is constrained by the presence of joints. We'll defer until a later section a full treatment of the rotational inertia of a rigid body. For now, a rigid body can be treated dynamically as a point particle, with all its mass at its center-of-mass and no rotational degrees-of-freedom (cf. 2.3).

We are assuming that the reader is generally familiar with Newton's three laws and their consequences.

2.2.1.1 Coordinate Systems

It is a general principle that the results of physical experiments should be independent of the coordinates with which they are expressed. In particular, we should be able to describe the motion of multiple bodies within a vehicle by any convenient choice of coordinates; not just by the global cartesian coordinates of each body. The first part of these notes makes use of three-dimensional cartesian vector coordinates $\vec{r} = (x, y, z)$ with respect to some fixed global inertial coordinate frame (cf. Section 2.2.1.3). This is shown in Figure 2.1. In later sections we will show how to expand our horizons, making use of the simplest "generalized" coordinates to express our equations of motion.

2.2.1.2 Newton's Second Law

If a "particle" has mass m and position vector $\vec{r}$ in a suitable coordinate system, then the first derivative with respect to time of its position vector is its *velocity* $\vec{v}$ (so that $\vec{v} = \dot{\vec{r}}$) and its second derivative is its *acceleration* $\vec{a}$ (thus $\vec{a} = \ddot{\vec{r}}$). The *linear momentum* of the

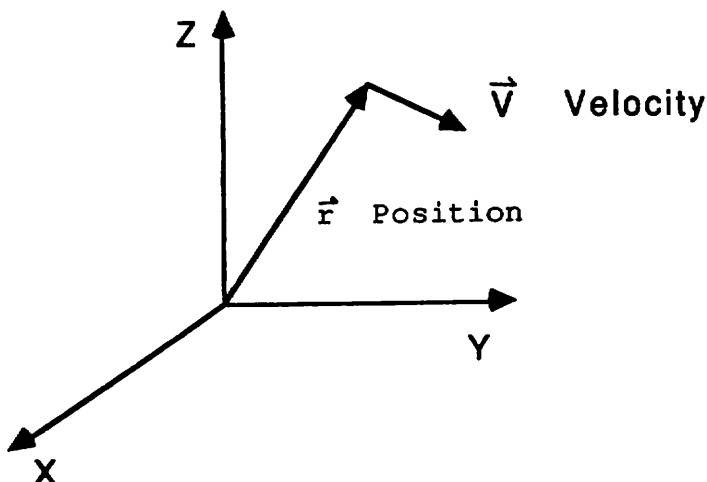


Figure 2.1: Cartesian coordinate system for a particle.

particle is the vector $\vec{p} = m\vec{v}$.

We now can present the only dynamics equation you will ever need: **Newton's Second Law** states that

$$\dot{\vec{p}} = \vec{F} \quad (2.1)$$

where $\vec{F}$ is the vector sum of all forces acting on the body. If the mass is constant with time (we assume this is always the case in these notes), this is usually stated as $m\vec{a} = \vec{F}$.

Everything we do is now downhill from this lofty summit. In essence, all other dynamics formulation are mere restatements of Newton's second equation; where the attempt is made to make it more convenient. For example, as we will see later, often it is very difficult to know the reaction forces which are generated by constrained motion. Newton's equation in the form of Equation 2.1 is then replaced by the more convenient virtual work equations.

Example of Newton's Second Law: Vertical motion of a mass m under gravitational acceleration g . The second law gives $m\ddot{z} = -mg$ (note the use

in two different ways of Newton's law - if we know the constant gravitational acceleration g , the gravitational *force* must be mg). Integrating this equation gives $z(t) = -gt^2/2 + \dot{z}_0 t + z_0$ where the initial conditions for velocity and position provide $\dot{z}_0$ and z_0 . If the mass has initial height $z = 0$ and velocity $\dot{z}_0 = v_0$ then $z(t) = -gt^2/2 + v_0 t$. Because Newton's equation is second-order, we must always prescribe both an initial position and velocity in order to solve the equation.

2.2.1.3 Inertial Reference Frames

We must remember that Newton's Second Law, Equation 2.1, is valid only in **inertial** reference frames. These are the special frames which are in uniform motion with respect to one-another. Only in these frames do the equations of dynamics become so simple. For our vehicle models we use an "inertial" Cartesian reference frame fixed to the earth¹. The final equations of vehicle motion are most conveniently written in terms of the earth's "inertial" frame, and not with respect to, say, a frame fixed on the vehicle chassis. Such a frame would not remain, in general, inertial, and the equations of motion would have additional funny force terms proportional to mass due to accelerations.

2.2.1.4 Newton's First Law

Notice that Newton's equation 2.1 tells us that if $\vec{F} = 0$, then we have $\vec{a} = 0$. Since $\vec{a} = \dot{\vec{v}}$, we see that $\vec{v} = \text{constant}$, and the body's momentum and velocity are constant. Thus we have **Newton's First Law** to the effect that *when forces are not acting, a body remains in uniform motion moving in a straight line*. The law of *conservation of linear momentum* states that in the absence of external forces, linear momentum is conserved (cf. the next paragraph). Linear momentum is then the tendency to continue moving in the same direction.

¹An alert reader will see that such a frame is not inertial due to the earth's rotation, but the effects are so small that to an engineering approximation we can treat it as inertial for a vehicle on a road.

2.2.1.5 Newton's Third Law

Newton gave yet another law to complete his dynamics. It is a statement about linear momentum conservation when internal forces acting along straight lines between bodies are present. Newton's Third Law state that *to every action, there is an equal and opposite reaction*².

Example of Newton's Third Law: A rocket moves forward because it ejects, due to chemical reactions generating heat, high-speed gases towards the rear. The rocket motion is the "reaction" to the "action" of the gas motion.

An important application of this law is to systems of particles. A system of particles without external forces acting cannot change its total linear momentum vector $\vec{P} = \sum_i \vec{p}_i$. This allows us to eliminate the (possible) internal forces between each pair of particles, and show that the motion of the total system acts as if all the system's mass was concentrated at a single point (the center of mass), which, when no external forces act, itself has a uniform motion.

Example of conservation of linear momentum: A bomb with mass m moving forward at velocity v explodes, the total system momentum must remain the same, so $\vec{P} = \sum_i \vec{p}_i$ remains constant even though individual bomb fragments have many different momenta.

2.2.1.6 Angular Momentum

We remember that the angular momentum $\vec{L}$ of a particle is defined as the "moment of liner momentum":

$$\vec{L} = \vec{r} \times \vec{p} \quad (2.2)$$

²This fails to be true in for some forces in electrodynamics since the magnetic Lorentz force does not act in this way.

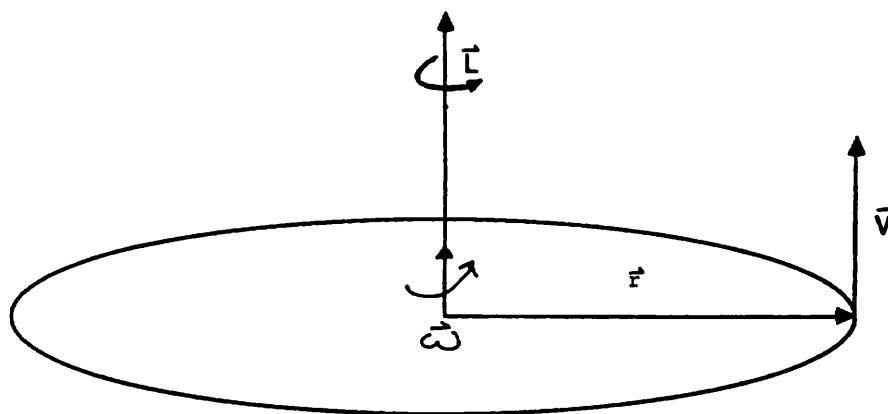


Figure 2.2: Angular momentum of particle moving in a circle.

where $\vec{p}$ is the particles *linear* momentum, and $\vec{r}$ is the position or radius vector from the fixed origin of coordinates to the particle. Angular momentum is intuitively considered as the tendency to continue rotating about a given point. In general, the angular momentum will clearly depend upon the choice of the center of coordinates. The torque $\vec{N}$ about the center of coordinates is the “moment of force $\vec{F}$ ” is defined to be

$$\vec{N} = \vec{r} \times \vec{F} \quad (2.3)$$

Example of angular momentum: A particle with mass m moving in a circle of radius r with uniform velocity v has an angular momentum of $L = rp = mrv$ about the center in the polar direction normal to the plane of motion. If ω is its angular velocity, then $v = r\omega$ and $L = mr^2\omega$. Figure 2.2 shown this example.

2.2.1.7 Rate of Change of Angular Momentum

If we now use Newton's second law, equation 2.1, and take the $\times$ -product of both sides with the radius vector $\vec{r}$, we get

$$\vec{N} = \vec{r} \times \dot{\vec{p}} \quad (2.4)$$

Using the definition of the angular momentum vector $\vec{L}$, this can be rewritten as

$$\vec{N} = \dot{\vec{L}} \quad (2.5)$$

We see that the rate of change of a particle's angular momentum is equal to the applied torque. If no torques are present, then the particle's angular momentum is constant. This is the law of **conservation of angular momentum**. In analogy with Newton's Third Law, for a system of particles, the total angular momentum $\vec{L} = \sum_i \vec{L}_i$ is conserved in the absence of external torques.

Example of conservation of angular momentum: For a central force, such as gravity, which acts on a line between the sun and a planet, the torque $\vec{N} = \vec{r} \times \vec{F} = 0$, and so the angular momentum of the *planet*, and not just the total sun-planet system, is conserved.

2.2.2 Multiple Particles and Constraints

As an introduction to multi-body dynamics, we first look at multi-particle systems. Since a set of free or interacting particles is not interesting for our ultimate application, we consider the existence of constraints restricting the free motion of a particle set.

2.2.2.1 Holonomic Constraints

A **constraint** is some extra condition between the particles' coordinates that limits the motion of the particles in some way.

Examples of constraints:

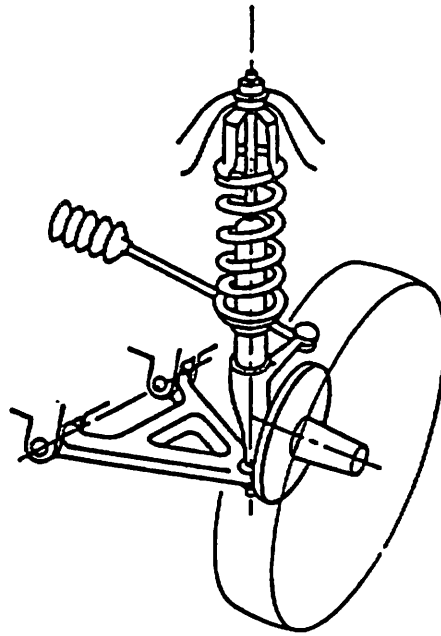


Figure 2.3: Constrained system - a Macpherson front suspension.

1. A bead on a wire is constrained to one dimensional motion by the wire.
2. A tire is constrained to revolve around the axis of its revolute joint.
3. A Macpherson front suspension is constrained to have only a vertical “jounce” and horizontal “steer” motion. The steer is itself constrained by the tie-rod to be a function of steering wheel position. This is shown in Figure 2.3.

A **holonomic constraint** is a explicit (possibly time dependent) relation among the particle coordinates $\vec{r}_i$ that can be expressed in the form

$$\Phi(\vec{r}_1, \vec{r}_2, \dots, t) = 0 \quad (2.6)$$

Other types of *non-holonomic* constraints exist (see [30]), but the holonomic ones are the most important, as well as the most pleasant to deal with in practice, since they reduce the dimensionality of the problem.

Examples of holonomic constraints: Various mechanical joint such as translational, revolute, and spherical joints all give holonomic constraints since they can all be expressed in terms of coordinate conditions, without using differentials. In addition, distance constraints are also holonomic.

1. Once a revolute joint axis $\vec{n}$ is specified, any motion of a vector $\vec{x}$ revolving about the joint must preserve the angle $\theta = \vec{n} \cdot \vec{x} / \|\vec{n}\| \|\vec{x}\|$. The holonomic constraint is then $\Phi = \vec{n} \cdot \vec{x} / \|\vec{n}\| \|\vec{x}\| - \cos(\theta) = 0$.
2. The existence of a tie-rod between steering rack and Macpherson suspension spindle enforces a holonomic distance constraint between the tie-rod attachment vectors.

Example of a non-holonomic constraint: In one dimension, if a tire rolls without slipping we have $x = r\theta$, so $d\Phi = dx - r d\theta$ is an integrable relation. See Figure 2.4. Thus we have a simple holonomic constraint, and we need only use either x or θ as the coordinate which describes the system. In two dimensions, the relation between position and angle for a tire rolling without slipping on a *plane* is not integrable. This constraint involves relations between the coordinate differentials. Let x, y be the coordinates of the center of the wheel, let θ be the angle the wheel has rotated, and let ϕ be the angle between the disk and the x axis. Then since the wheel rolls without slipping, the center velocity must be related to rotation rate by $v = r\dot{\theta}$ and the components of center motion are

$$\begin{aligned}\dot{x} &= v \sin(\phi) \\ \dot{y} &= -v \cos(\phi)\end{aligned}$$

These give the differential relations between the variables

$$\begin{aligned}dx - r d\phi \sin(\phi) &= 0 \\ dy + r d\phi \cos(\phi) &= 0\end{aligned}$$

It is not possible to integrate this differential relation to get a holonomic constraint between tire hub coordinates and tire rotation angle. The non-holonomic constraint gives conditions between the differentials which must be satisfied, without reducing the number of coordinates required to describe the motion.

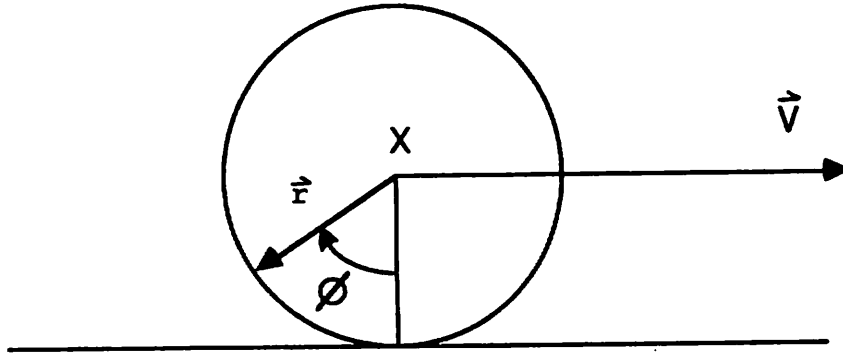


Figure 2.4: Constrained system - wheel rolling without slipping.

2.2.2.2 Multiple Constraints

Let us assume we have M particles each with coordinates $\vec{r}_i$. Often such particles are not free and subject only to known forces, but also move under the influence of reaction forces provided by the constraints. The existence of reaction forces are what force the system to obey the k *constraint equations* given by some large set of holonomic constraints such as

$$\begin{aligned}
 \Phi_1(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_M, t) &= 0 \\
 \Phi_2(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_M, t) &= 0 \\
 &\vdots \\
 \Phi_k(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_M, t) &= 0
 \end{aligned}
 \tag{2.7}$$

For notational purposes, it is easier to write the Cartesian coordinates of all the bodies as the components of one large vector $\vec{x}$ of length N ,

where

$$\begin{aligned}
 x_1 &= \vec{r}_{1,x} \\
 x_2 &= \vec{r}_{1,y} \\
 x_3 &= \vec{r}_{1,z} \\
 x_4 &= \vec{r}_{2,x} \\
 &\vdots \\
 x_N &= \vec{r}_{M,z}
 \end{aligned} \tag{2.8}$$

Using this notation we can consider the above set as k components of a *vector* constraint equation given by

$$\vec{\Phi}(\vec{x}) = 0 \tag{2.9}$$

2.2.2.3 Constraint Jacobian

The set of holonomic constraint conditions determines some maximal amount of dependence among the coordinates $\vec{r}_i$. The amount of dependence is measured by the row-rank of the **constraint Jacobian matrix** $\vec{\Phi}_{\vec{x}}$ whose i,j -th component is

$$\vec{\Phi}_{\vec{x},i,j} = \left(\frac{\partial \vec{\Phi}_i}{\partial x_j} \right) \tag{2.10}$$

where $1 \leq i \leq k$ and $1 \leq j \leq N$. The row-rank is the number of linearly independent rows of the matrix, and is thus the dimension of the image space for the associated linear transformation. Something special happens when the row rank is as large as possible; that is, when it is equal to the number k of constraint equations. This occurs when all the constraints are independent and thus by the implicit function theorem³, we can split our original coordinates $\vec{x}$ into a of $N-k$ independent coordinates and a set of k dependent coordinates. Thus, after renumbering the coordinates, we have

$$\begin{aligned}
 x_1 &= f_1(x_{N-k}, x_{N-k+1}, \dots) \\
 x_2 &= f_2(x_{N-k}, x_{N-k+1}, \dots) \\
 &\vdots \\
 x_k &= f_k(x_{N-k}, x_{N-k+1}, \dots)
 \end{aligned} \tag{2.11}$$

³See any good advanced calculus text or look in [37].

This is how independent constraints can be used to eliminate coordinate variables. We will see this again when we discuss generalized coordinates.

An effectively computable method of finding the independent coordinates is to take the LU-decomposition or, better yet, the **singular value decomposition**⁴ of the Jacobian.

Example of a constraint Jacobian of two linked particles: Consider two particles moving on the plane $z = 0$, which are also linked rigidly together at unit distance. We have $\vec{r}_1, \vec{r}_2$ as position vectors. Thus we have six coordinates x_i . Since the particles are the plane $x_3 = z_1 = 0$ and $x_6 = z_2 = 0$. Since distance between them is constant, we have $(\vec{r}_1 - \vec{r}_2)^2 = 1$. Then the constraint Jacobian is given by

$$\vec{\Phi}_{\vec{x}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 2x_1 & 2x_2 & 0 & -2x_4 & -2x_5 & 0 \end{pmatrix} \quad (2.12)$$

which clearly has the maximal row rank of three since both particles cannot simultaneously be at the origin.

In order to see why the Jacobian enters, we take equation 2.9 and form its differential for each component i :

$$\sum_j \frac{\partial \vec{\Phi}_i}{\partial x_j} dx_j = 0 \quad (2.13)$$

This is just a set of linear equations in the dx_j 's with coefficients given by the partials derivatives of Φ_i , in other words, by the Jacobian matrix. The maximal set of independent differentials, and so of coordinates, is then given by the row rank of the Jacobian matrix.

2.2.3 Principle of Virtual Work

In this section, the subscript i with denote an individual particle with position vector $\vec{r}_i$. When the position vectors of a system of particles is

⁴See the reference [29].

considered, we write $\vec{r} = (\vec{r}_1, \vec{r}_2, \dots)$. This convention will be followed for the remainder of the section.

2.2.3.1 Statement of Virtual Work Principle

The Principle of Virtual Work states that *for a dynamical system to be in equilibrium, the sum of the internal and external virtual work must vanish*. Clearly, this deserves some elaboration. What is being said is that a mechanical system is in its equilibrium state if when, holding time fixed, we vary any of the coordinates infinitesimally in a way compatible with all constraints, no work is performed. For static equilibrium, only the external forces have to be included in the virtual work since the reaction forces from the constraints are in general workless. If dynamic equilibrium is wanted, we must include the work due to inertial forces.

2.2.3.2 Relation with other Variational Methods

We remark that the virtual work formulation of dynamics is the most general available. When only conservative forces are present⁵, then for static equilibrium, it is equivalent to the minimization of the system's *potential energy* $E = \phi$ ⁶. But no such potentials exist in general for *dissipative* forces such as viscous friction. An even more fundamental problem exists if the forces of constraint can themselves do work - as in the case of rolling friction. The total system energy is now *total energy = internal energy - external work*. If this is minimized, we recover our original virtual work principle.

In general, for complex mechanical systems with dissipative elements, the virtual work principle seems in practice to be superior

⁵If a force can be written as the gradient of a potential function $\vec{f} = \nabla\phi$, work done by the force does not depend on the path taken in configuration space, only the starting and ending positions matter. In this case the force is called conservative. The work done by a non-conservative force, such as friction, clearly does depend upon the path. If we move a particle against friction and then return to the starting point, work has been done.

⁶The kinetic energy of a particle is $T = p^2/2m$, and it measures the energy of motion. Potential energy is a function only of position, e.g. height in a gravitational field. Only the sum of these two energies is conserved when no work is performed on or by the system.

to other less general variational techniques. Powerful methods due to Lagrange and Hamilton unfortunately are currently unsuitable for vehicle simulation.

2.2.3.3 Derivation of Virtual Work Principle

A virtual (infinitesimal) displacement of a system refers to a change in its configuration as the result of any arbitrary infinitesimal change of coordinates $\delta \vec{r}_i$, *consistent with the forces and constraints imposed on the system at time t* . The term virtual is used to distinguish these displacements from real kinematic displacements during the time dt when the forces and constraints may be changing.

Suppose that the system is in equilibrium so that the total force on each particle vanishes $\vec{F}_i = 0$. Then the scalar product

$$\vec{F}_i \cdot \delta \vec{r}_i \quad (2.14)$$

which is the virtual work of the force $\vec{F}_i$ in the displacement $\delta \vec{r}_i$, also vanishes. Hence the sum over all bodies

$$\sum_i \vec{F}_i \cdot \delta \vec{r}_i = 0 \quad (2.15)$$

which so far says nothing new.

If we now decompose the force into its external component $\vec{F}_i^{ext}$, and internal (constraint) component $\vec{F}_i^{int}$,

$$\vec{F}_i = \vec{F}_i^{ext} + \vec{F}_i^{int} \quad (2.16)$$

we get that

$$\sum_i (\vec{F}_i^{ext} \cdot \delta \vec{r}_i + \vec{F}_i^{int} \cdot \delta \vec{r}_i) = 0 \quad (2.17)$$

Suppose we now restrict ourselves to systems where the *net virtual work of the forces of constraint is zero*⁷. Then our equation 2.16 becomes

$$\sum_i \vec{F}_i^{ext} \cdot \delta \vec{r}_i = 0 \quad (2.18)$$

⁷This is true for rigid bodies and will be assumed for all systems considered from now on. It can fail if, for example, rolling friction is present.

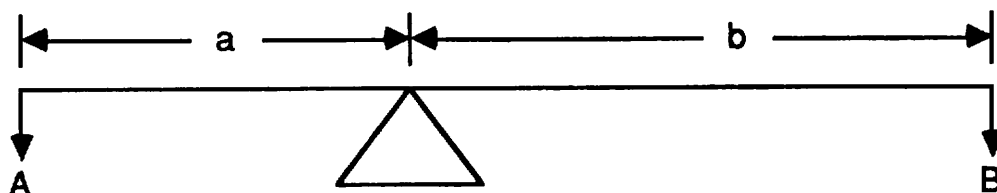


Figure 2.5: Virtual work principle - the lever.

The equation 2.18 is the *static* principle of virtual work.

Example of static principle of virtual work: Consider a lever with one arm length a and the other b as shown in Figure 2.5. External forces A and B are applied at each end. Let x and y be the heights of each end. Since the lever is rigid and so has only one degree-of-freedom, we have a constraint between x and y such that $x/a = y/b$. The principle of virtual work states that $A\delta x + B\delta y = 0$ for kinematically admissible displacements, i.e. ones for which $\delta y = -b\delta x/a$. Thus the principle gives that $(A - bB/a)\delta x = 0$ for every virtual displacement δx and so the condition for equilibrium of the lever is $Aa = Bb$, i.e. the moments must balance. Note that if we did this problem using Newton's second law, we would have to explicitly include the reaction force from the fulcrum, in addition to the external forces A and B .

2.2.3.4 D'Alembert's Principle

In order to obtain a principle which is valid for more general motion, we use a device that originated with J. Bernoulli and developed by D'Alembert. Start with the equation of motion equivalent to 2.1:

$$\vec{F}_i = \dot{\vec{p}}_i \quad (2.19)$$

This can be written

$$\vec{F}_i - \dot{\vec{p}}_i = 0 \quad (2.20)$$

which can be interpreted as saying that the particle is in equilibrium with the total effective force $\vec{F}_i - \dot{\vec{p}}_i$ made up of external and inertial parts. Thus instead of 2.18, we have⁸

$$\sum_i (\vec{F}_i^{ext} - \dot{\vec{p}}_i) \cdot \delta \vec{r}_i = 0 \quad (2.21)$$

This is called **D'Alembert's principle**.

Example of dynamic virtual work principle: Consider a pendulum of mass m supported by a chord of length r . See Figure 2.6. It has gravity as its only external force. The virtual work equation is $\delta \vec{x}^T (m\ddot{\vec{x}} + mg\hat{z}) = 0$. But $\vec{x} = (r \sin(\theta), r(1 - \cos(\theta)))$ where θ is the angle the chord makes with the vertical. By the chain-rule, $\delta \vec{x} = (r \cos(\theta), -r \sin(\theta)) \delta \theta$ and our virtual work equations becomes $\delta \theta (\ddot{\theta} + g \sin(\theta)) = 0$. Since the virtual rotation $\delta \theta$ is arbitrary, the dynamic equation of motion for the pendulum is $\ddot{\theta} + (g/r) \sin(\theta) = 0$. This is the classical equation for the pendulum. Notice that any dependence on the mass has disappeared. It can be solved exactly in terms of elliptic functions, or approximated by the harmonic oscillator equation for small θ .

2.2.3.5 Kinematically Admissible Virtual Displacements

A virtual displacement is **kinematically admissible** when it is consistent with the constraint conditions $\vec{\Phi} = 0$. By this we mean that it

⁸ Again, we are assuming that the forces of constraint produce no virtual work.

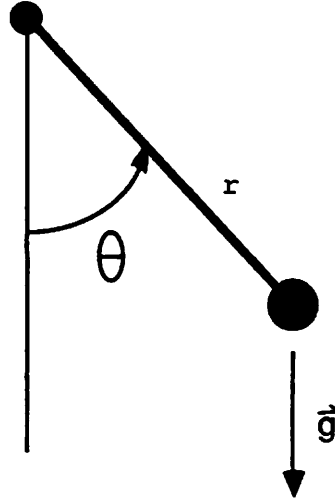


Figure 2.6: Virtual work principle - the pendulum.

satisfies the variational form of the constraint condition

$$\vec{\Phi}_{\vec{r}} \delta \vec{r} = 0 \quad (2.22)$$

The equation 2.22 follows directly from the chain rule since δ acts like differentiation⁹.

We can interpret equation 2.22 geometrically as follows: The constraint equation defines a hypersurface in phase space¹⁰. At each point of the surface, there is a tangent hyperplane. The rows of the constraint Jacobian form a basis which spans a *complementary subspace* to the tangent space. For example, if we have a space with N coordinates and the row rank of the constraint Jacobian is rk , then the tangent space is of dimension $N - rk$. Thus the row rank of $\vec{\Phi}_{\vec{r}}$ determines the dimensionality of the hyperplane, and hence the constraint surface. Equation 2.22 simply says that the kinematically admissible displacements are those that are orthogonal to this com-

⁹This can be shown by taking Taylor's expansion of $\vec{\Phi}(\vec{r} + \delta \vec{r})$ at $\vec{r}$ to first order.

¹⁰A way of stating that the solution to an equation will define some higher dimensional surface in the coordinate space.

plementary space. Thus the admissible displacements are those living in the tangent hyperplane.

A good example is when we have a single constraint equation $\Phi(\vec{r}) = 0$. In this case, the constraint Jacobian is just the transpose of the constraint's gradient $\nabla\Phi$. But this gradient is always perpendicular to the surface¹¹ and the vectors which are perpendicular to it are by definition the elements of the surfaces tangent plane at that point. So the kinematically admissible displacements are the elements of the surfaces tangent hyperplane.

2.2.3.6 An Essential Difficulty

We must now make a crucial observation: We would like to take Equation 2.21 and claim that each term in the sum must vanish individually. This cannot be done as the equation stands because Equation 2.21 is in terms of the underlying Cartesian virtual displacement's $\delta\vec{r}_i$ which must still satisfy *all admissible kinematic constraint conditions* the system is subjected to¹². We cannot conclude because the total vector sum must vanish, or even that each individual term must vanish, that for each i

$$(\vec{F}_i^{ext} - \dot{\vec{p}}_i) \cdot \delta\vec{r}_i = 0 \quad (2.23)$$

We *could* do this if the individual $\delta\vec{r}_i$ were without constraint, since then they are arbitrary, not having to satisfy and admissibility conditions. Imposition of constraints on the system's motion produces relations between the various components of each $\delta\vec{r}_i$ which destroy the required independence. A very fancy way to say this is that the principle of virtual work requires only that the vectors

$$\vec{F}_i^{ext} - \dot{\vec{p}}_i$$

¹¹This is the general property of gradients. ∇f is always perpendicular to any level surface $f = \text{const.}$

¹²In geometric terms, they are only elements of the lower dimensional tangent spaces to the constraint surface.

be orthogonal to the subspace of virtual displacements that are admissible, and so, in geometric terms, are elements of the constraint surface tangent space.

The way around this problem is to introduce either generalized coordinates which have the constraints built in and so the virtual change in the coordinates are independent, or to introduce what are called **Lagrange multipliers** which also allow for independent virtual displacements, but at the cost of adding an extra unknown vector. We now review generalized coordinates and then discuss the variational equations of motion.

2.2.4 Generalized Coordinates

Sometimes it is easy to pick a new set of coordinates that are more natural to the problem at hand¹³ and have *all* or *most* of the system constraints already built into them. These are then the **generalized coordinates**¹⁴ for the system.

Examples of generalized coordinate systems:

1. Pendulum - angle between attachment chord and vertical.
2. Planetary motion about the sun - polar coordinates in plane of orbit.
3. Macpherson front suspension - vertical translational "jounce" and steer angle of spindle.

What we require for generalized coordinates is that we can write the Cartesian coordinates for each body uniquely in terms of the generalized coordinates $q_1, q_2, \dots, q_n$. Thus the time variation of the generalized coordinates completely determines the dynamics. The minimal number of generalized coordinates needed in describing the system is called the **degrees of freedom**¹⁵ of the system. We will

¹³By this we mean they may considerably simplify the problems formulation.

¹⁴Which need not have the dimensions of length

¹⁵Written often as DOF's.

see later how to know when one has a correct set of generalized coordinates given a set of constraint equations, but it should be clear to the reader that generalized coordinates must number at least $N - \text{rk}(\tilde{\Phi}_q)^{16}$, so that enough coordinates are available to describe the motion. If more are given, then they are not independent, and must satisfy constraint conditions.

Examples of counting DOF's:

- One DOF:
 1. A piston moving up and down. A translational joint.
 2. A rigid body rotating about a fixed axis. A revolute joint.
- Two DOF:
 1. A particle moving on a surface.
- Three DOF:
 1. A particle moving in space.
 2. A rigid body rotating about a fixed point (a top).
 3. A spherical joint.
- Four DOF:
 1. Two components of a double star revolving in the same plane.
- Five DOF:
 1. Two particles at a constant distance from each other.
- Six DOF:
 1. Two planets revolving about the sun.
 2. A rigid body moving freely in space.

Example of constrained particle cases:

¹⁶Where $\text{rk}(M)$ is the rank of the matrix M . A theorem in linear algebra states that row rank = column rank = rank of a matrix.

1. If a particle were constrained to move on a curve, we would need only a single generalized coordinate.
2. If, for example, the particles were constrained to move on a plane, there are fewer independent coordinates needed to completely describe the system. Each particle now requires only two coordinates, so the total system requires $2M$. We have imposed M independent constraints on the particles of the system. Thus only $2M$ generalized coordinates would be required to fully describe the motion.
3. Suppose in the general case we have M particles and have imposed k constraints. Then as in section 2.2.2 we would intuitively think that we need only $q_1, q_2, \dots, q_{3M-k}$ generalized coordinates if the Jacobian has maximum row rank, where we have

$$\begin{aligned}
 \vec{r}_1 &= \vec{r}_i(q_1, q_2, \dots, q_{3M-k}, t) \\
 \vec{r}_2 &= \vec{r}_i(q_1, q_2, \dots, q_{3M-k}, t) \\
 &\vdots \\
 \vec{r}_M &= \vec{r}_i(q_1, q_2, \dots, q_{3M-k}, t)
 \end{aligned} \tag{2.24}$$

The number $3M-k$ is the total number of *degrees of freedom* for the system.

2.2.4.1 Change of Variables to Generalized Coordinates

Suppose we have our M bodies described by a set of N generalized coordinates. We always assume that the relation Equation 2.24 is differentiable, so that we can relate the derivatives of the two systems of coordinates. This is done using the chain rule and the **Jacobian of the coordinate transformation**. The Jacobian is the matrix $\vec{r}_q$ of partial derivatives

$$\vec{r}_q = \begin{pmatrix} \frac{\partial \vec{r}_{1,x}}{\partial q_1} & \dots & \frac{\partial \vec{r}_{1,x}}{\partial q_N} \\ \frac{\partial \vec{r}_{1,y}}{\partial q_1} & \dots & \frac{\partial \vec{r}_{1,y}}{\partial q_N} \\ \frac{\partial \vec{r}_{1,z}}{\partial q_1} & \dots & \frac{\partial \vec{r}_{1,z}}{\partial q_N} \\ \vdots & \vdots & \vdots \\ \frac{\partial \vec{r}_{M,x}}{\partial q_1} & \dots & \frac{\partial \vec{r}_{M,x}}{\partial q_N} \end{pmatrix} \tag{2.25}$$

Example of a single particle Jacobian: If we have only one particle whose coordinate vector is $\vec{r}$, then its Jacobian matrix with respect to the m generalized coordinates $q_1, q_2, \dots, q_m$ will be

$$\vec{r}_{q,i,j} = \left(\frac{\partial \vec{r}_i}{\partial q_j} \right) \quad (2.26)$$

where $1 \leq i \leq 3$ and $1 \leq j \leq m$.

2.2.4.2 Chain Rule

We need the general version of the chain rule. We just state the result and leave the proof to the references. The chain rule tells you how to change variables when you are differentiating. Suppose $\vec{y} = \vec{y}(\vec{x})$, then¹⁷

$$\begin{pmatrix} \frac{\partial}{\partial y_1} \\ \frac{\partial}{\partial y_2} \\ \vdots \\ \frac{\partial}{\partial y_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_m} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \dots & \frac{\partial y_n}{\partial x_m} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_m} \end{pmatrix} \quad (2.27)$$

or, in much neater matrix notation,

$$\frac{\partial}{\partial \vec{y}} = \vec{y}_x \frac{\partial}{\partial \vec{x}} \quad (2.28)$$

As a special case, we have the time derivative of a vector with respect to the time derivatives of the new coordinates

$$\dot{\vec{y}} = \vec{y}_x \dot{\vec{x}} \quad (2.29)$$

Example of a particle on a curve: A position on a space curve is determined by a single generalized coordinate q ¹⁸. Then the particles Cartesian position $\vec{r}$ can be written as

$$\vec{r} = \vec{r}(q) \quad (2.30)$$

¹⁷We use operator notation in the next equation, since the operand is a dummy place-holder anyway.

¹⁸Which could be taken as arclength along the curve.

After differentiating, the particle's velocity is, using the chain rule, just

$$\dot{\vec{r}} = \vec{r}_q \dot{q} \quad (2.31)$$

where the Jacobian $\vec{r}_q$ is the curves **tangent vector**. The particles acceleration vector is then

$$\ddot{\vec{r}} = \vec{r}_q \ddot{q} + \vec{r}_{qq} \dot{q}^2 \quad (2.32)$$

The first term on the right can be interpreted as the generalized acceleration $\ddot{q}$ and the second term as a centripetal acceleration $\dot{q}^2$. This is obvious in the case of uniform circular motion, since then $\ddot{q} = 0$.

Example of a particle on a surface: In this case we have two generalized coordinates $\vec{q} = [q_1, q_2]^T$, and the coordinates are $\vec{r} = \vec{r}(\vec{q})$. The velocity relation is now

$$\dot{\vec{r}} = \vec{r}_{\vec{q}} \dot{\vec{q}} \quad (2.33)$$

and acceleration

$$\ddot{\vec{r}} = \vec{r}_{\vec{q}} \ddot{\vec{q}} + (\vec{r}_{\vec{q}\vec{q}} \dot{\vec{q}}) \dot{\vec{q}} \quad (2.34)$$

Again, the first term is the generalized acceleration, while the second term, which is the Jacobian of a vector derived from the Jacobian, is a combination of mysterious accelerations such as the coriolis and centripetal. The form of this equation remains the same for any general set of coordinates.

Example of an application to holonomic constraints: If we take a holonomic constraint and differentiate, we get

$$\vec{\Phi}_{\vec{r}} \dot{\vec{r}} = -\dot{\Phi}_t \quad (2.35)$$

which gives a condition that the velocities $\dot{\vec{r}}$ must satisfy in order to be consistent with the constraint. If the constraint is time independent, then the right hand side of the equation will vanish, and so the constraint Jacobian will annihilate the velocity vector, that is to say, the vector is in the **null space** or **kernel** of the Jacobian matrix.

If we differentiate the constraint one more time, then we have

$$\vec{\Phi}_{\vec{r}} \ddot{\vec{r}} = -\gamma \quad (2.36)$$

where

$$\gamma = (\vec{\Phi}_{\vec{r}\vec{r}} \dot{\vec{r}}) \dot{\vec{r}} + 2\vec{\Phi}_{\vec{r}t} \dot{\vec{r}} + \vec{\Phi}_{tt} \quad (2.37)$$

If we work instead with generalized coordinates, we have

$$\vec{\Phi}_{\vec{q}} = \vec{\Phi}_r \vec{r}_{\vec{q}} \quad (2.38)$$

and the equations 2.35 and 2.36 become

$$\begin{aligned} \vec{\Phi}_{\vec{q}} \dot{\vec{q}} &= -\Phi_t \\ \vec{\Phi}_{\vec{q}} \ddot{\vec{q}} &= -\gamma \end{aligned} \quad (2.39)$$

where now

$$\gamma = (\vec{\Phi}_{\vec{q}} \ddot{\vec{q}})_{\vec{q}} \dot{\vec{q}} + 2\vec{\Phi}_{\vec{q}t} + \Phi_{tt} \quad (2.40)$$

2.2.4.3 Kinematically Admissible Generalized Virtual Displacements

In the same way as section 2.2.3.5, we have kinematically admissible **generalized** virtual displacements $\delta \vec{q}$. Again, by the chain rule we have

$$\vec{\Phi}_{\vec{q}} \delta \vec{q} = 0 \quad (2.41)$$

where now $\vec{\Phi}_{\vec{q}} = \vec{\Phi}_r \vec{r}_{\vec{q}}$.

2.2.4.4 D'Alembert's Principle in Generalized Coordinates

We now transform the virtual work Equation 2.21 to generalized coordinates and forces.

First, we consider a single body with generalized coordinates $\vec{q}$. From $\vec{r} = \vec{r}(\vec{q})$, we have the generalized virtual displacement

$$\delta \vec{r} = \vec{r}_{\vec{q}} \delta \vec{q} \quad (2.42)$$

and the time derivatives

$$\dot{\vec{r}} = \vec{r}_{\vec{q}} \dot{\vec{q}} \quad (2.43)$$

and acceleration

$$\ddot{\vec{r}} = \vec{r}_{\vec{q}} \ddot{\vec{q}} + (\vec{r}_{\vec{q}} \dot{\vec{q}})_{\vec{q}} \dot{\vec{q}} \quad (2.44)$$

We then can write equation 2.21 in terms of all kinematically admissible $\delta \vec{q}$ as

$$m \delta \vec{q}^T \ddot{\vec{r}}_{\vec{q}} + (\ddot{\vec{r}}_{\vec{q}} \vec{q})_{\vec{q}} \dot{\vec{q}} - \delta W^{ext} = 0 \quad (2.45)$$

where δW^{ext} is the external virtual work given by

$$\delta W^{ext} = \delta \vec{q}^T \ddot{\vec{r}}_{\vec{q}} \vec{F}^{ext} \quad (2.46)$$

The quantity $\vec{Q}^{ext} = \ddot{\vec{r}}_{\vec{q}}^T \vec{F}^{ext}$ is called the **generalized external force**. After collecting terms, we can write equation 2.45 as

$$\delta \vec{q}^T (m \ddot{\vec{r}}_{\vec{q}} \ddot{\vec{q}} + m \ddot{\vec{r}}_{\vec{q}} (\ddot{\vec{r}}_{\vec{q}} \vec{q})_{\vec{q}} \dot{\vec{q}} - \vec{Q}^{ext}) = 0 \quad (2.47)$$

for all kinematically admissible virtual generalized displacements.

Thus if we have an independent set of generalized coordinates, all virtual displacements are kinematically admissible and we must have

$$m \ddot{\vec{r}}_{\vec{q}} \ddot{\vec{q}} + m \ddot{\vec{r}}_{\vec{q}} (\ddot{\vec{r}}_{\vec{q}} \vec{q})_{\vec{q}} \dot{\vec{q}} - \vec{Q}^{ext} = 0 \quad (2.48)$$

Note that by picking independent generalized coordinates, we have solved our essential difficulty! This is because we are no longer restricted to some subspace of virtual displacements which must be orthogonal to the vector

$$m \ddot{\vec{r}}_{\vec{q}} \ddot{\vec{q}} + m \ddot{\vec{r}}_{\vec{q}} (\ddot{\vec{r}}_{\vec{q}} \vec{q})_{\vec{q}} \dot{\vec{q}} - \vec{Q}^{ext} \quad (2.49)$$

but now *every* vector in the space of virtual displacements must be orthogonal to 2.49 and so the vector must vanish.

The case of multiple bodies is identical. One just must add an index i to each of the quantities above. The crucial observation is that one may choose an independent set of generalized coordinates. This gives us

$$\sum_i m_i \ddot{\vec{r}}_{i\vec{q}} \ddot{\vec{q}} - \vec{Q}^{ext} + \sum_i m_i \ddot{\vec{r}}_{i\vec{q}} (\ddot{\vec{r}}_{i\vec{q}} \vec{q})_{\vec{q}} \dot{\vec{q}} = 0 \quad (2.50)$$

where the summation is over the number of bodies, and the external generalized force is

$$\vec{Q}^{ext} = \sum_i \ddot{\vec{r}}_{i\vec{q}}^T \vec{F}_i^{ext} \quad (2.51)$$

2.2.5 Lagrange Multipliers

In this section, we show an alternative way of dealing with deriving the equations of motion with respect to constrained coordinates. With generalized coordinates, we eliminated the constraints by choosing a proper set of new independent coordinates. With Lagrange multipliers, the constraints are dealt with directly, and are incorporated into the equations of motion.

2.2.5.1 Lagrange Multiplier Theorem

Haug in reference [37] calls the following theorem the **Lagrange multiplier theorem**. Suppose we have $\vec{b}$ in R^n and A a $m \times n$ matrix. If

$$\vec{s}^T \vec{b} = 0 \quad (2.52)$$

for all $\vec{s}$ in R^n such that

$$A\vec{s} = 0 \quad (2.53)$$

then there exists a vector $\vec{\lambda}$ in R^m , called a **Lagrange multiplier**, such that

$$\vec{s}^T \vec{b} + \vec{s}^T A^T \vec{\lambda} = 0 \quad (2.54)$$

for all $\vec{s}$ in R^n , and if A has full row rank, then $\vec{\lambda}$ is unique.

It is worth looking at this from a geometric point of view. We claim that it says something fairly interesting geometrically. The first condition, Equation 2.52, can be interpreted as saying $\vec{b}$ is orthogonal to the subspace of vectors that A takes to zero (equation 2.53), $\vec{b} \perp \ker(A)$ ¹⁹, and the conclusion, equation 2.54, is that $\vec{b}$ is the image of some vector $-\vec{\lambda}$ under the matrix transformation A^T . This last statement can be written $\vec{b} \in \text{im}(A^T)$. That either this is true or the first condition must then be false, and there is a vector with $\vec{s}^T \vec{b} \neq 0$, is called the **Fredholm alternative theorem**. It follows essentially from the fact that if a vector is in the kernel of a transformation, there is no way to recover it as the inverse image under the transformation of some other vector.

¹⁹As we have said before, the set of vectors $\vec{s}$ such that $A\vec{s} = 0$ is called the *kernel* of A .

2.2.5.2 Application of the Theorem

Again, we will restrict ourselves to a single particle with coordinates $\vec{r}$ for simplicity.

We know that

$$\delta \vec{r}^T (m \ddot{\vec{r}} - \vec{F}) = 0 \quad (2.55)$$

for all admissible virtual displacements $\delta \vec{r}$ that satisfy

$$\vec{\Phi}_{\vec{r}} \delta \vec{r} = 0 \quad (2.56)$$

Now, in the Multiplier theorem, take

$$\vec{b} = m \ddot{\vec{r}} - \vec{F}$$

and

$$A = \vec{\Phi}_{\vec{r}}$$

Then we must have a $\vec{\lambda}$ such that

$$\delta \vec{r}^T (m \ddot{\vec{r}} - \vec{F} + \vec{\Phi}_{\vec{r}}^T \vec{\lambda}) = 0 \quad (2.57)$$

for all $\delta \vec{r}$. We have done what we wanted to, we now have an equation that must hold for every virtual displacement, and so the coefficient of $\delta \vec{r}$ must vanish. Our **Lagrange Multiplier form of the constrained Cartesian dynamics equations** are then

$$(m \ddot{\vec{r}} - \vec{F} + \vec{\Phi}_{\vec{r}}^T \vec{\lambda}) = 0 \quad (2.58)$$

It will be very important for numerical applications to note that equation 2.58 is *not* an ordinary differential equation, but, because of $\vec{\lambda}$, is instead an *differential-algebraic* equation. These demand special numerical techniques for their solution.

For multiple particles, we write

$$\vec{r} = [\vec{r}_1^T, \vec{r}_2^T, \dots, \vec{r}_N^T]^T$$

and

$$\vec{F} = [\vec{F}_1^{extT}, \vec{F}_2^{extT}, \dots, \vec{F}_N^{extT}]^T$$

for the particle position vectors and external forces. The mass matrix is then the diagonal matrix $M = \text{diag}(m_1 I, \dots, m_N I)^{20}$. The multiple particle version of equation 2.58 is then

$$(M\ddot{\vec{r}} - \vec{F} + \vec{\Phi}_r^T \vec{\lambda}) = 0 \quad (2.59)$$

In the generalized coordinate formulation using Lagrange multipliers²¹, we use the Jacobian as our coordinate transformation. If we proceed as above, but use the chain rule for taking derivatives with respect to our generalized coordinates, we get the dynamical equations

$$\vec{r}_q^T M \ddot{\vec{r}}_q \ddot{\vec{q}} + \vec{r}_q^T M (\ddot{\vec{r}}_q \dot{\vec{q}}) \dot{\vec{q}} - \vec{Q}^{ext} + \vec{\Phi}_q^T \vec{\lambda} = 0 \quad (2.60)$$

2.2.5.3 Matrix Formulation of Constrained Equations of Motion

We can combine all our equations into one nice matrix equation. By combining equations 2.36 and 2.59, we get

$$\begin{pmatrix} M & \vec{\Phi}_r^T \\ \vec{\Phi}_r & 0 \end{pmatrix} \begin{pmatrix} \ddot{\vec{r}} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{F}^{ext} \\ \vec{\gamma} \end{pmatrix} \quad (2.61)$$

Haug in reference [37] shows that this equation, when combined with the obvious proper initial conditions, provides a complete description of the system dynamics as long as M is positive definite for all kinematically admissible vectors²²

Similarly, we can form the matrix version of the generalized coordinate constrained dynamics equations using equations 2.39 and 2.60. First, it's useful to rewrite equation 2.60 as

$$\hat{M} \ddot{\vec{q}} + \vec{\Phi}_q^T \vec{\lambda} = \vec{Q}^{ext} - \vec{r}_q^T M (\ddot{\vec{r}}_q \dot{\vec{q}}) \dot{\vec{q}} \quad (2.62)$$

where we have used the reduced mass matrix

$$\hat{M} = \vec{r}_q^T M \vec{r}_q \quad (2.63)$$

²⁰Where I is the identity matrix of the proper dimension.

²¹We can indeed choose generalized coordinates that themselves are subject to additional constraints.

²²A matrix M is positive definite if for all vectors $\vec{a}$, $\vec{a}^T M \vec{a} \geq 0$. This is equivalent to having only positive eigenvalues.

We also define a new *force* that combines our generalized force with the *pseudo-force* terms that arise from the non-inertial reference frames of the generalized coordinates

$$\vec{R} = \vec{Q}^{ext} - \vec{r}_{\vec{q}}^T M(\vec{r}_{\vec{q}} \dot{\vec{q}}) \dot{\vec{q}} \quad (2.64)$$

We then have as our matrix equation

$$\begin{pmatrix} \hat{M} & \vec{\Phi}_{\vec{q}}^T \\ \vec{\Phi}_{\vec{q}} & 0 \end{pmatrix} \begin{pmatrix} \ddot{\vec{q}} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{R} \\ \vec{\gamma} \end{pmatrix} \quad (2.65)$$

Haug [37] shows that this is a unique prescription for the system's dynamics whenever the generalized coordinates are non-singular and the constraint Jacobian has full row-rank.

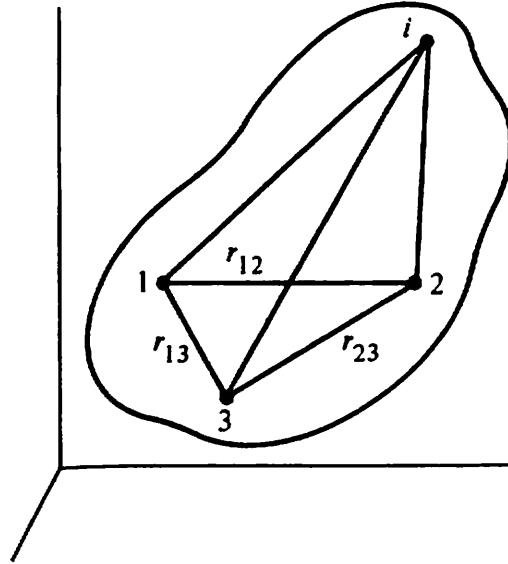


Figure 2.7: Rigid Body.

2.3 Rigid Body Kinematics

This section reviews rigid body kinematics. It introduces both Euler angles and Euler parameters. In particular, it gives the description of rigid body motion required by the vehicle dynamics model.

2.3.1 The Rigid Body

We describe some of the essential aspects of rigid bodies.

2.3.1.1 Definition of a Rigid Body

We think of “solid” bodies as objects rigid in their shape, with each point of the body holding a fixed position with respect to all the other points. More precisely, a **rigid body** is a large collection (thought of as infinite in number) of particles labeled by their position vectors $\vec{r}_i$, and which have a constant distance $c_{i,j} = |\vec{r}_i - \vec{r}_j|^2$ between each pair of particles. See Figure 2.7. As the rigid body undergoes acceleration, reaction forces must exist in order to hold the rigid body together,

satisfying the holonomic constraint conditions. The constraint conditions of fixed distances between all points insures that all angles between particles are also fixed.

A rigid body is the first approximation to most engineering structures, when we can ignore the compliance of the system, and treat it a perfectly rigid object. Such an object has no internal energy²³ since the internal constraints do no work. If internal energy is important for the problem at hand, such as when the energy stored in a spring is a significant part of the total system energy, then the rigid approximation fails. The present dynamics models treat many parts of the vehicle as rigid bodies, since the elastic energy for those parts is of no importance to the total system. Thus the strain energy of, say, the front Swiss arm is probably not crucial to the dynamics, but the coil springs must be treated as compliant bodies.

We remember that a rigid body is fully described by *six* independent coordinates or degrees of freedom; we must specify the position of a single point within the body (usually its center of mass. Cf. the next section) and its orientation. Both the point and the orientation require three parameters to be determined in space. We'll quickly review the center of mass, and then take some time thinking about the idea of how a body's orientation is best described.

2.3.1.2 Center of Mass

The center of mass is the position $\vec{R}$ at which all the inertial moments vanish, and would be the point (not necessarily in the body) at which if it were supported there, it would not rotate if the support were accelerated. We can calculate the center of mass $\vec{R}$ for a set of particles as

$$\vec{R} = \frac{\sum m_i \vec{r}_i}{\sum m_i} \quad (2.66)$$

and, for a solid rigid body, as

$$\vec{R} = \frac{\int m(\vec{x}) \vec{r}(\vec{x})}{\int m(\vec{x})} \quad (2.67)$$

where the integrals are taken over the solid's volume.

²³It has, for example, no elastic or plastic strain energy.

The center of mass turns out to be a particularly good place to take as the origin of coordinates in many dynamics problems.

Examples of centers of masses:

1. The center-of-mass of a disk with uniform density is at its geometric center.
2. A front engined car chassis with engine has a center-of-mass forward of its geometric center due to the mass of the engine. The exact c.g. shifts under vehicle loading.

2.3.1.3 Rigid Body Orientation

The angular orientation of the rigid body is measured by how an orthogonal coordinate system *fixed in the body*²⁴ changes its angular relationship to a orthogonal coordinate system *fixed in the global inertial system*²⁵ (see 2.8). We can measure this by the **direction cosines** of the body fixed coordinates with respect to the world coordinates. If we call the unit vectors in the $\vec{x}'$ - system

$$\vec{i}', \vec{j}', \vec{k}'$$

and in the $\vec{x}$ -system

$$\vec{i}, \vec{j}, \vec{k}$$

²⁴We will denote this as the $\vec{x}'$ -system.

²⁵Denoted as the $\vec{x}$ -system.

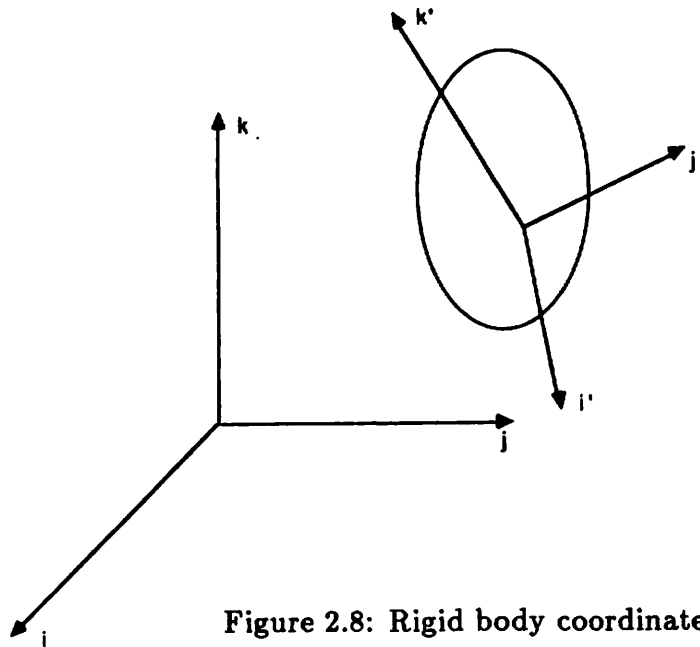


Figure 2.8: Rigid body coordinates.

then the nine direction cosines are given by

$$\begin{aligned}
 \alpha_1 &= \vec{i}' \cdot \vec{i} \\
 \alpha_2 &= \vec{i}' \cdot \vec{j} \\
 \alpha_3 &= \vec{i}' \cdot \vec{k} \\
 \beta_1 &= \vec{j}' \cdot \vec{i} \\
 \beta_2 &= \vec{j}' \cdot \vec{j} \\
 \beta_3 &= \vec{j}' \cdot \vec{k} \\
 \gamma_1 &= \vec{k}' \cdot \vec{i} \\
 \gamma_2 &= \vec{k}' \cdot \vec{j} \\
 \gamma_3 &= \vec{k}' \cdot \vec{k}
 \end{aligned} \tag{2.68}$$

These are called direction *cosines* since the dot product of unit vectors is just the cosine of the angle between them. The direction cosines are the components of the $\vec{x}'$ unit vectors with respect to the $\vec{x}$ unit vectors.

Because each of the set of coordinate unit vectors are *orthogonal*, that is, they are pairwise at right angles with one another for our Cartesian

coordinate systems and are of unit length²⁶, they form the components of an orthogonal matrix A ²⁷

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix} \quad (2.69)$$

We could describe the orientation of the body by the direction cosines, but we would need nine parameters connected by the six conditions of forming an orthogonal matrix. This would be very inefficient, but could be done. Instead, we will try to find a smaller set of parameters that will also do the job, but first we review rotations in Cartesian three-space.

2.3.2 The Description of Rotations

We now describe the theory of rotations in three-space. This can be made quite abstract, but here I only present a utility grade theory. The most important fact about rotations is that you *can't* add them in the same way as you would vectors. They are different beasts²⁸, and only infinitesimal rotations are true vectors in space. Only the

²⁶These conditions are of the form $\vec{i} \cdot \vec{j} = 0$ and $\vec{k} \cdot \vec{k} = 1$, etc.

²⁷An *orthogonal* matrix is one whose transpose is its inverse, so $A^T = A^{-1}$. This is equivalent to the columns being orthogonal unit vectors.

²⁸They are in fact elements of a non-commutative Lie group of real 3×3 orthogonal matrices, with unit determinants, called $SO(3)$. They naturally form a smooth manifold which can be locally parameterized with three coordinates in a manner such that all group operations are also smooth functions of the parameters. The Lie group $SO(3)$ is not simply connected, but has another group called $SU(2)$ of complex 2×2 unitary matrices with unit determinant which is its simply connected covering space. The Lie algebra associated with $SU(2)$ are the 2 Hermitian matrices. The real form of these, the real 3×3 skew-symmetric matrices form the Lie algebra of $SO(3)$. The infinitesimal rotations are the members of this Lie algebra of $SO(3)$, and so can be associated with 3-dimensional vectors. In real terms, $SU(2)$ is the unit sphere in real 4-space. The standard real parameterization of this sphere will be the four Euler parameters e_0, e_1, e_2, e_3 with the normalization $e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$ that I will introduce in a much more elementary manner in Section 2.3.2.4 (a complex parameterization by two Cayley-Klein parameters is also sometimes used). The Lie algebras of each map to their corresponding group via the matrix exponential. A basis for the Lie algebra of $SU(2)$ is the identity matrix and the three Pauli spin matrices σ_i . The two Lie algebras can be identified by the mapping $\vec{x} \mapsto \vec{x} \cdot \vec{\sigma}$,

definition of Euler angles and Euler parameters, along with the finite and infinitesimal rotation formulas are critical.

2.3.2.1 Rotations as Orthogonal Matrices

The reader should be familiar with the fact that the orthogonal matrix

$$M_z(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.70)$$

is a counter-clockwise (right-handed) rotation about the z-axis of θ radians. Similarly, we can write down rotations about the y-axis

$$M_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (2.71)$$

and x-axis

$$M_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix} \quad (2.72)$$

A crucial (and complicating property) of rotations is that the order in which you perform finite rotations matters²⁹! The reader is urged to check that a rotation of a body about the x-axis followed by a rotation about the y-axis is not in general the same total rotation as if we first rotated about the y-axis and then rotated about the x-axis. $M_z M_y \neq M_y M_z$, so M_z and M_y do not *commute*.

2.3.2.2 Euler Angles

As noted, the nine direction cosines are not a good set of orientation coordinates since they are subject to six relations. To find the minimal independent set (by dimension counting, there are three since the six relations are independent) of three parameters, we break the

while the map between $SU(2)$ and $SO(3)$ is a 2-fold covering transformation and so half-angles will start to appear in the formulas.

²⁹ We will see later that for an *infinitesimal* rotation order does not matter since they are true vectors, and so are commutative in their addition.

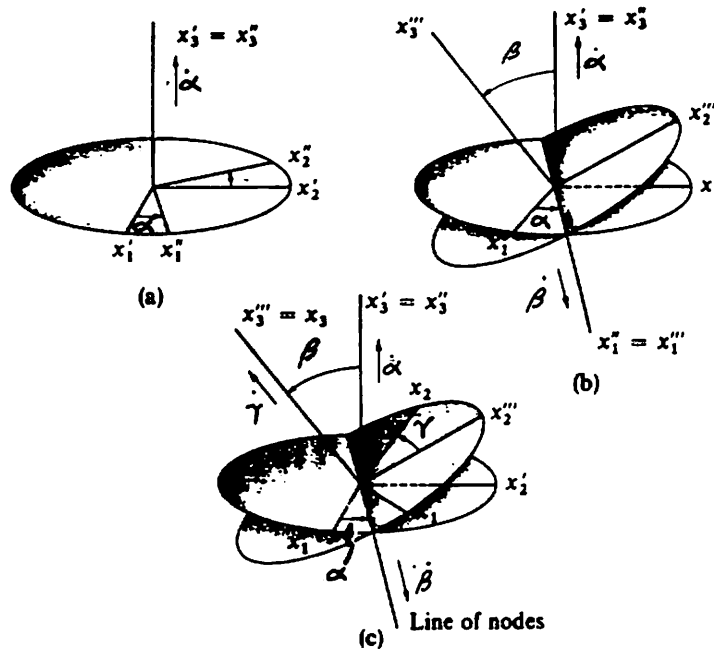


Figure 2.9: Diagram illustrating the three successive rotations through Euler angles α , β , and γ .

angular motion into three rotations about special axes. These will be the Euler angles.

An arbitrary orientation of the coordinate axis (and so of the rigid body) can be obtained by the following sequence of successive rotations about single axes³⁰

1. Rotating (xyz) through an angle α about the z -axis to give $(x'y'z')$. The orthogonal matrix is given by $M_z(\alpha)$.
2. Rotating $(x'y'z')$ through an angle β about the y' -axis to give $(x''y''z'')$. The orthogonal matrix is given by $M_{y'}(\beta)$.
3. Rotating $(x''y''z'')$ through an angle γ about the z'' -axis to give $(x'''y'''z''')$. The orthogonal matrix is given by $M_{z''}(\gamma)$.

This is illustrated in Figure 2.9.

³⁰Remember, the order in which we perform the rotations matters!

The complete orthogonal transformation is

$$M(\alpha, \beta, \gamma) = M_z(\alpha)M_y(\beta)M_z(\gamma) \quad (2.73)$$

and so $M(\alpha, \beta, \gamma) =$

$$\begin{pmatrix} \cos(\beta) \cos(\alpha) \cos(\gamma) & \cos(\beta) \sin(\alpha) \cos(\gamma) & -\sin(\beta) \cos(\gamma) \\ -\sin(\alpha) \sin(\gamma) & \cos(\alpha) \sin(\gamma) & \\ -\cos(\beta) \cos(\alpha) \sin(\gamma) & -\cos(\beta) \sin(\alpha) \sin(\gamma) & \sin(\beta) \sin(\gamma) \\ -\sin(\alpha) \cos(\gamma) & \cos(\alpha) \cos(\gamma) & \\ \sin(\beta) \cos(\alpha) & \sin(\beta) \sin(\alpha) & \cos(\beta) \end{pmatrix} \quad (2.74)$$

We must warn the reader that there is nothing special about which axes the rotations are to take place. There are other conventions in use. But the sequence given above is the standard modern physics one. An alternative that occurs in engineering is to use a *yaw*, *pitch*, and *roll* sequence.

The Euler angles are a minimal set of rotation coordinates. They can be used to find orientation from Euler angle rates since they are additive. But the reader is encouraged to check that the transformation Jacobian of Equation 2.74 is *singular* at certain values of the parameters. thus it is not a good set of coordinates for numerical calculations, since extra care would be involved in avoiding the singular nature of the coordinates. A better set of coordinates are the *Euler parameters*. But first, we motivate their existence by deriving what is called the finite rotation formula.

2.3.2.3 Finite Rotation Formula

Suppose we are given a unit vector $\vec{n}$, another vector $\vec{r}$, and ask for the new vector $\vec{r}'$ which arises when we rotate $\vec{r}$ through an angle Φ about the axis vector $\vec{n}$. The answer is elementary and can be obtained from geometry by considering Figure 2.10. Then

$$\begin{aligned} \vec{r}' &= O\vec{N} + N\vec{V} + V\vec{Q} \\ &= \vec{n}(\vec{n} \cdot \vec{r}) + (\vec{r} - \vec{n}(\vec{n} \cdot \vec{r})) \cos(\Phi) + (\vec{n} \times \vec{r}) \sin(\Phi) \end{aligned} \quad (2.75)$$

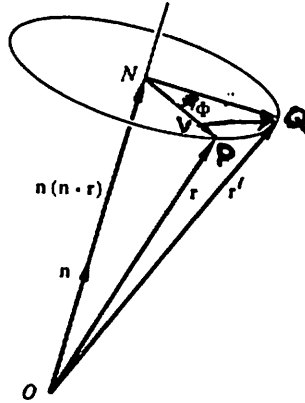


Figure 2.10: Derivation of finite rotation formula.

This gives

$$\vec{r}' = \vec{r} \cos(\Phi) + \vec{n}(\vec{n} \cdot \vec{r})(1 - \cos(\Phi)) + (\vec{n} \times \vec{r}) \sin(\Phi) \quad (2.76)$$

This formula works for any magnitude of rotation.

2.3.2.4 Euler Parameters

If we start from the finite rotation formula Equation 2.76, there is a magic parameterization of it in terms of four **Euler parameters** $e_0, \vec{e} = (e_1, e_2, e_3)$. These are related to a finite rotation by angle χ about an axis unit vector $\vec{n}$ by

$$\begin{aligned} e_0 &= \cos(\chi/2) \\ \vec{e} &= \vec{n} \sin(\chi/2) \end{aligned} \quad (2.77)$$

Euler parameters clearly satisfying a normalization condition $e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$.

Then from trigonometric identities,

$$\cos(\Phi) = e_0^2 - e_1^2 - e_2^2 - e_3^2 \quad (2.78)$$

and

$$\vec{\eta} \sin(\Phi) = 2e_0 \vec{e} \quad (2.79)$$

Equation 2.76 and then be written

$$\vec{r}' = \vec{r}(e_0^2 - e_1^2 - e_2^2 - e_3^2) + 2\vec{e}(\vec{e} \cdot \vec{r} + 2(\vec{e} \times \vec{r})e_0) \quad (2.80)$$

If we now expand Equation 2.80 component-by-component, we can recover the cosine matrix M which relates $\vec{r}'$ to $\vec{r}$. It is

$$M = \begin{pmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1e_2 - e_0e_3) & 2(e_1e_3 + e_0e_2) \\ 2(e_1e_2 + e_0e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2e_3 - e_0e_1) \\ 2(e_1e_3 + e_0e_2) & 2(e_2e_3 - e_0e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{pmatrix} \quad (2.81)$$

If we define

$$E = \begin{pmatrix} -e_1 & e_0 & -e_3 & e_2 \\ e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{pmatrix} \quad (2.82)$$

$$G = \begin{pmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{pmatrix} \quad (2.83)$$

then

$$M = EG^T \quad (2.84)$$

We can then relate the Euler angles to the Euler parameters by the formulas

$$\begin{aligned} e_0 &= \cos\left(\frac{\phi + \psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \\ e_1 &= \cos\left(\frac{\phi - \psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \\ e_2 &= \sin\left(\frac{\phi - \psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \\ e_3 &= \sin\left(\frac{\phi + \psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \end{aligned} \quad (2.85)$$

The explicit form of this construction is very important, since it provides a way of parameterizing rotations in a non-singular manner.

The reader is encouraged to check that the Jacobian of the mapping is everywhere of full rank as promised. It is in terms of the Euler parameters that the numerical solution of rigid body motion can be performed most readily since the parameters are always good and so provide a stable method of describing the motion.

2.3.2.5 Infinitesimal Rotations

Suppose if instead of a finite angle Φ , the rotation is infinitesimal in the sense that it is so small we can always ignore second- or higher-order terms in any Taylor series expansion involving Φ . Thus if we make the approximations $\cos(\Phi) \approx 1$ and $\sin(\Phi) \approx \Phi$, Equation 2.76 becomes

$$\vec{r}' = \vec{r} + \Phi(\vec{\eta} \times \vec{r}) \quad (2.86)$$

We can use Equation 2.86 to write the effect of a infinitesimal rotation on a vector $\vec{r}$ in the concise form

$$\begin{aligned} M(\Phi)\vec{r} &= I\vec{r} + \Phi\vec{\eta} \times \vec{r} \\ &= (I + \Phi\vec{\eta} \times)\vec{r} \\ &= (I - \vec{\Phi} \times)\vec{r} \end{aligned} \quad (2.87)$$

where $\vec{\Phi} = \Phi\vec{\eta}$.

A useful device is to notice that the operation of "taking the cross product" $\vec{a} \times$ is the same as multiplication by the skew-symmetric matrix $\tilde{a}$ generated by the vector $\vec{a}$

$$\tilde{a} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \quad (2.88)$$

It is easy to check that $\vec{a} \times \vec{x} = \tilde{a}\vec{x}$. In particular, we can replace $\Phi\vec{\eta} \times$ by $\vec{\Phi} \times$.

2.3.3 Rate of Change of a Vector

In this section, we find the kinematic description of velocity and acceleration for a rigid body rotating with respect to a fixed inertial frame. Rotation angles will be denoted θ , and the angular velocity $\vec{\omega}$.

2.3.3.1 Infinitesimal Rotational Relations

From equation 2.86, we see that for a small rotation $\tilde{\theta}$, we can write the rotation's affect on a vector $\vec{r}$ as

$$M(\theta)\vec{r} = \vec{r} + \tilde{\theta}\vec{r} \quad (2.89)$$

Thus the infinitesimal change $d\vec{r}$ is

$$\begin{aligned} d\vec{r} &= M(\theta)\vec{r} - \vec{r} \\ &= \tilde{\theta}\vec{r} \end{aligned} \quad (2.90)$$

Because we want to require that $\tilde{\theta}$ be infinitesimal, we'll write it $d\tilde{\theta} = \tilde{\omega}^{31}$, so

$$d\vec{r} = \tilde{d}\theta\vec{r} \quad (2.91)$$

2.3.3.2 Vector Rate of Change Equation

We see that the rate of change, or velocity, due to a rotation alone, is

$$d\vec{r}/dt = \tilde{\omega}\vec{r} \quad (2.92)$$

It should be clear to the reader that the total rate of change of any vector *observed in a fixed inertial reference frame*, is the total of the change *as observed in a frame fixed in the body* and the change due to the relative rotation rate $\tilde{\omega}$ of the two frames. Thus the total rate of change of any vector $\vec{G}$

$$(d\vec{G}/dt)_{space} = (d\vec{G}/dt)_{body} + \tilde{\omega}\vec{G} \quad (2.93)$$

2.3.3.3 Velocity and Acceleration

If we now take $\vec{G}$ to be the position vector of a point on the rigid body, we have

$$(d\vec{r}/dt)_{space} = (d\vec{r}/dt)_{body} + \tilde{\omega}\vec{r} \quad (2.94)$$

and so in terms of the velocities

$$\vec{v}_{space} = \vec{v}_{body} + \tilde{\omega}\vec{r} \quad (2.95)$$

³¹Remember, the infinitesimal rotations, in contrast to finite rotations, $\tilde{\omega}$ really are *vectors*, and so can be added.

If we apply our equation 2.93 to $\vec{v}_{space}$, we get a famous and important result relating the accelerations in the two frames

$$\begin{aligned} (d\vec{v}_{space}/dt)_{space} &= \vec{a}_{space} \\ &= (d\vec{v}_{space}/dt)_{body} + \tilde{\omega}\vec{r}_{space} \\ &= \vec{a}_{body} + 2(\tilde{\omega}\vec{v}_{body}) + \tilde{\omega}\tilde{\omega}\vec{r} \end{aligned} \quad (2.96)$$

Equations 2.95 and 2.96 are our fundamental kinematic equations for rigid bodies.

The two “extra” terms in the right-hand side of equation 2.96 are of special interest and should be known to the reader. They exist because of the non-inertial rotating frame. The term

$$\vec{a}_{coriolis} = 2(\tilde{\omega}\vec{v}_{body}) \quad (2.97)$$

is called the **Coriolis acceleration**, and h

$$\vec{a}_{centrifugal} = \tilde{\omega}\tilde{\omega}\vec{r} \quad (2.98)$$

is called the **Centrifugal acceleration**. The Coriolis force exists only when motion is occurring with respect to the rotating frame. The angular velocity enters as the first power, while the centrifugal force depends upon the angular velocity to the second power.

2.3.4 Euler's and Chasles' Theorems

Two theorems about rigid bodies are relatively easy to establish by considering the eigenvalues of an orthogonal matrix. Any 3×3 orthogonal matrix has only one real eigenvalue $\lambda = 1$, the other two are complex conjugates with non-zero imaginary part. The eigenvector corresponding to $\lambda = 1$ is an axis vector for the rotation, while the other's are complex.

Euler's theorem on the motion of a rigid body states that

Theorem 1 (Euler) *The general displacement of a rigid body with one point fixed is a rotation about some axis.*

Because any vector along the axis of rotation is left fixed, it must be an eigenvector of the transformation, with a unit eigenvalue. The proof of Euler's theorem consists of showing such an eigenvalue exists for any transformation with a fixed point.

A corollary of Euler's theorem is called **Chasles' theorem** which states that

Theorem 2 (Chasles) *The most general displacement of a rigid body is a translation plus a rotation.*

These are the final things we have to say about rigid body *kinematics*.

2.4 Dynamics of Rigid Bodies

In this section, we review the inertia matrix and present the Newton-Euler equations of motion for a rigid body. A theorem already mentioned due to Chasles' states that an arbitrary motion of a rigid body can be represented by a translation plus a rotation. This suggests that we can decompose the dynamics of a rigid body into two independent parts: Translation and rotation. The translation part is easy and just involves the center-of-mass motion. The rotational part involves the concept of angular momentum and the inertia matrix or tensor.

2.4.1 Rigid Body Motion

We can treat a rigid body as a collection of particles held together by some unknown (and arbitrary) set of internal forces which act along a line between the particles in an "opposite and equal" fashion. We know that the translational and angular momentum are conserved in the absence of external forces and torques, and if, indeed, such forces and torques are present, we have the dynamical equations

$$\vec{F}^{ext} = \dot{\vec{p}} \quad (2.99)$$

and

$$\vec{N}^{ext} = \dot{\vec{L}} \quad (2.100)$$

We have also previously remarked that the angular momentum depends upon the choice of origin of coordinates.

If we have two coordinate systems attached to the rigid body as shown in Figure 2.11, then an essential property of the body is that all points move together, translating and rotating at the same rate. Thus we have both angular velocity vectors for the two coordinate systems being equal and arrive at the conclusion that the angular velocity vector of a rigid body, *with respect to any coordinate system*, are equal³². We can prove the following relationships

$$\vec{v}_2 = \vec{v}_1 + \vec{\omega} \vec{R} \quad (2.101)$$

$$\vec{\omega}_2 = \vec{\omega}_1 \quad (2.102)$$

³²The proof of this is easy, and can be found in [30].

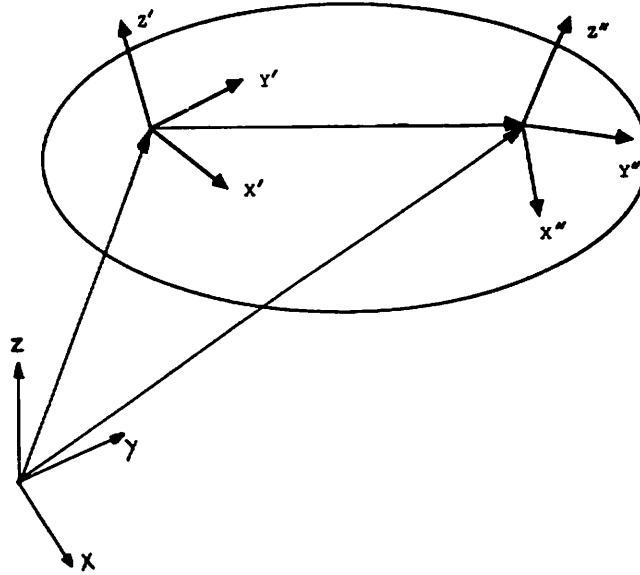


Figure 2.11: Relation between sets of rigid body coordinates with different origins.

Note that translational motion does not have this absolute relationship. This relation implies, when combined with Equation 2.95, that at any instant, a new center of coordinates can always be chosen so, with respect to this new coordinate system, the body undergoes a *pure rotational motion*, without any translation.

2.4.2 Angular Momentum

The total angular momentum $\vec{L}$ of a rigid body is

$$\vec{L} = \int_V (\vec{r} \times \vec{p}) dV \quad (2.103)$$

Its value depends upon from where $\vec{r}$ is measured, that is, from the choice of the origin. If we pick a second origin, so that $\vec{r} = \vec{r}' + \vec{a}$, then

$$\vec{L} = \vec{L}' + \vec{a} \times \vec{p} \quad (2.104)$$

We see that the angular momentum depends on the choice of origin, except when the system is at rest as a whole, so that $\vec{p} = 0$. If we now

take as our coordinate system the center-of-mass frame, then we can replace $\vec{a} = \vec{R}$.

2.4.3 Inertia Tensor

If we consider how to express the kinetic energy of a rigid body, we find that there should be a *linear* relation between the body's angular momentum and its rotational velocity. We can write it as

$$\vec{L} = J\vec{\omega} \quad (2.105)$$

where J is the inertia matrix (or tensor since it is a doubly-indexed quantity $J_{i,j}$ with the proper transformation property for a second rank covariant tensor).

We can write the inertia tensor as

$$J_{i,j} = \int_V \rho(\vec{r})(r^2\delta_{i,j} - x_i x_j) dV \quad (2.106)$$

where ρ is the body's mass density. An important, yet obvious property of the inertia tensor, is that it is *symmetric*. From this, it follows that all its eigenvalues are real with orthogonal eigenvectors (if the eigenvalues are distinct the eigenvectors are orthogonal, otherwise we can choose an orthogonal basis for the degenerate eigenspace) and that by choosing a new coordinate system called the **principal axes** given by the three orthogonal eigenvectors, we can diagonalize the inertia matrix J , so that only the diagonal **principal moments of inertia** remain, with all the off-diagonal **products of inertia** vanishing. For this reason, we will almost always choose the local body coordinate frame to align with the principal axes of inertia. For a body with finite volume, one can show that in fact J is positive-definite, as well as symmetric. This ensures that all eigenvalues are positive and that our final equations of motion will have a unique solution.

Examples of the Inertia Tensor:

1. As a first example, consider a spherical mass of uniform density. Then from symmetry, the only non-vanishing elements of J are

$$J_{1,1} = J/3$$

$$\begin{aligned} J_{2,2} &= J/3 \\ J_{3,3} &= J/3 \end{aligned} \quad (2.107)$$

where

$$J = 2 \int_V \rho(\mathbf{r}) r^2 dV \quad (2.108)$$

2. A second example is the *symmetrical top*, where the body has a rotational symmetry about some axis. Since we are free to choose the coordinates so that this axis is the z-axis, we have

$$\begin{aligned} J_{1,1} &= J_{2,2} \\ &= (1/2) \int_V \rho(\vec{r})(r^2 + z^2) dV \\ J_{3,3} &= \int_V \rho(\vec{r})(x^2 + y^2) dV \end{aligned} \quad (2.109)$$

A large collection of examples can be found in [45].

2.4.3.1 Newton-Euler Equations

The motion of a rigid body can be decomposed into two parts: The translational motion of its center-of-mass, treated as a particle with the total body mass, and purely rotational motion about the center-of-mass. (A proof of this can be found in any mechanics text, e.g. [30].)

If $\vec{r}$ is the body's center-of-mass position in some inertial frame and m is its total mass, the equation of motion for $\vec{r}$ is just Newton's

$$m\ddot{\vec{r}} = \vec{f} \quad (2.110)$$

where $\vec{f}$ are total external forces acting on the body. Suppose $\vec{L}$ is the body's angular momentum. Then we know that

$$\vec{L} = J\vec{\omega} \quad (2.111)$$

where I is the moment of inertia of the body and $\vec{\omega}$ is the body's angular velocity. This relation is true in any coordinate frame, in particular, in a frame fixed with the body. But Newton's equation

for the rate of change for angular momentum expressed with vector components in the inertial frame gives

$$\dot{\vec{L}} = \vec{n} \quad (2.112)$$

($\vec{n}$ the applied torque) when combined with our rate of change formula

$$\dot{\vec{L}} = \dot{\vec{L}}_{body} + \tilde{\omega} \vec{L} \quad (2.113)$$

expressed in body-fixed coordinates gives **Euler's Equation**

$$J' \dot{\vec{\omega}}' + \tilde{\omega}' J' \vec{\omega}' = \vec{n}' \quad (2.114)$$

since the inertia matrix J' is *constant in the body frame*. The difficulty with using this equation is that although J' may simple in the body-fixed frame (if we choose principal axes), the angular velocity vector is more complex. Often it is most convenient to express the external torques in terms of the local body frame. Equation 2.114 in component form is

$$\begin{aligned} J'_1 \dot{\omega}'_1 - \omega'_2 \omega'_3 (J'_2 - J'_3) &= n'_1 \\ J'_2 \dot{\omega}'_2 - \omega'_3 \omega'_1 (J'_3 - J'_1) &= n'_2 \\ J'_3 \dot{\omega}'_3 - \omega'_1 \omega'_2 (J'_1 - J'_2) &= n'_3 \end{aligned} \quad (2.115)$$

where the vectors are expressed in the principal axis frame of the body.

We can transform Equation 2.114 to global coordinates by making use of the cosine matrix A between the local body frame and the global frame. In this case, A is itself a function of the position of the body parameters, and $\dot{A}$ is not zero,

$$J \dot{\vec{\omega}} - J \dot{A} A^T \vec{\omega} + \tilde{\omega} J \vec{\omega} = \vec{n} \quad (2.116)$$

Example of torque-free motion of a rigid body: If there are no torques applied to the body, the angular momentum is conserved and the equations

for the angular velocity in the body-fixed principal axis frame are Equations 2.115

$$\begin{aligned} J_1 \dot{\omega}_1 &= \omega_2 \omega_3 (J_2 - J_3) \\ J_2 \dot{\omega}_2 &= \omega_3 \omega_1 (J_3 - J_1) \\ J_3 \dot{\omega}_3 &= \omega_1 \omega_2 (J_1 - J_2) \end{aligned} \quad (2.117)$$

We know that rotational kinetic energy and total angular momentum are conserved and so they are “integrals of the motion”. Using this fact, these equations can be explicitly integrated in terms of elliptic functions. There is also an interesting geometric interpretation of the motion due to Poinsot that can be found in [30], where the body’s motion can be reduced to the rolling of its inertia ellipsoid on the invariable plane whose normal vector is the fixed angular momentum $\vec{L}$. During this rolling, the path traced out on the ellipsoid is called the polhode, while the corresponding path on the plane is called the herpolhode. Equations 2.117 are equivalent to the wonderful statement that “the polhode rolls without slipping on the herpolhode lying in the invariable plane”.

If we assume a symmetric top, where

$$J_1 = J_2 = J_{12} \neq J_3,$$

then Equations 2.115 become

$$\begin{aligned} J_{12} \dot{\omega}_1 &= \omega_2 \omega_3 (J_{12} - J_3) \\ J_{12} \dot{\omega}_2 &= \omega_3 \omega_1 (J_3 - J_{12}) \\ J_3 \dot{\omega}_3 &= 0 \end{aligned} \quad (2.118)$$

Then $\omega_3 = \text{constant}$ and the first two equations become

$$\begin{aligned} \dot{\omega}_1 &= -\frac{J_3 - J_{12}}{J_{12}} \omega_3 \omega_2 \\ \dot{\omega}_2 &= \frac{J_3 - J_{12}}{J_{12}} \omega_3 \omega_1 \end{aligned} \quad (2.119)$$

If we define $\Omega = \frac{J_3 - J_{12}}{J_{12}} \omega_3$, then we have

$$\begin{aligned} \dot{\omega}_1 + \Omega \omega_2 &= 0 \\ \dot{\omega}_2 - \Omega \omega_1 &= 0 \end{aligned} \quad (2.120)$$

which has the well known solution

$$\begin{aligned} \omega_1 &= A \cos(\Omega t) \\ \omega_2 &= A \sin(\Omega t) \end{aligned}$$

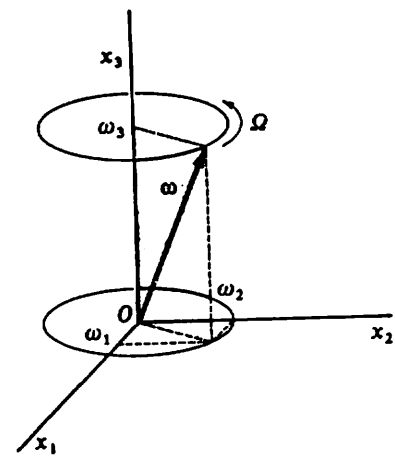


Figure 2.12: Symmetric top.

The vector ω then has constant length and the angular velocity vector ω in the body fixed frame rotates in the x', y' -plane with angular velocity Ω . In the global frame, this corresponds to precession of the z' -axis. See Figure 2.12 Equations 2.110 and 2.114 can be combined into a single matrix

equation

$$M\dot{\vec{Y}} = \vec{F} \quad (2.121)$$

where

$$M = \begin{pmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & J' \end{pmatrix} \quad (2.122)$$

is symmetric and positive-definite,

$$\vec{Y} = \begin{pmatrix} \vec{r} \\ \vec{\omega}' \end{pmatrix} \quad (2.123)$$

$$\vec{F} = \begin{pmatrix} \vec{f} \\ \vec{n}' - \vec{\omega}' J' \vec{\omega}' \end{pmatrix} \quad (2.124)$$

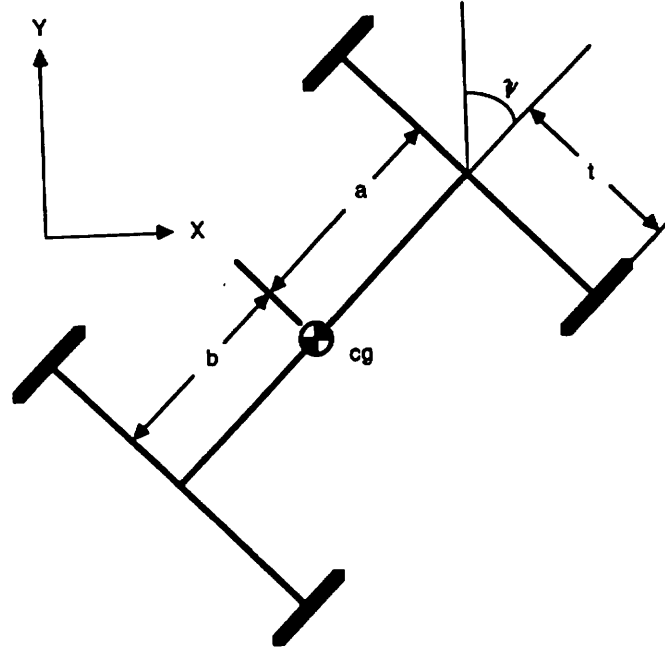


Figure 2.13: Two degree-of-freedom car model.

The combined equations 2.121 for the translational and rotational motion of a rigid body is often called the **Newton-Euler Equation**. It is a first order equation in the translational and angular velocities $\vec{v}$ and $\vec{\omega}$. To calculate the body position and orientation, we add the coupled first-order system for translation

$$\begin{aligned}\dot{\vec{r}} &= \vec{v} \\ m\dot{\vec{v}} &= \vec{f}\end{aligned}\tag{2.125}$$

Example of 2 degree-of-freedom car model: Consider the car coordinates shown in Figure 2.13. We assume it is traveling forward at a constant velocity U and with small angular deviations. Let x, y be the chassis c.g. coordinates. The front axle is a distance a in front of the c.g. and the rear axle a distance b to the rear. The tires are attached at a distance t from the centerline. The coordinates of the front wheels are

$$\begin{aligned}x_{1,2} &= x + a \cos(\psi) \pm t \sin(\psi) \\ &= x + a \pm t\psi \\ y_{1,2} &= y + a \sin(\psi) \mp t \cos(\psi)\end{aligned}$$

$$= x + a\psi \mp t$$

The sideslip angle α , which is the angle between the tire hub velocity vector and the tire heading, is $\alpha = \dot{y}/\dot{x} - \psi$. Lateral complicated functions of normal force and slip angle. We assume that for small slip angles, the lateral forces are linear functions of the slip angle, with c_f, c_r the initial slope of the lateral force/sideslip curve at a fixed normal force. Since we have assumed that $\dot{x} = U$, the slip angles for the front tires are $\alpha_{1,2} = \alpha_f = (\dot{y} + a\dot{\psi})/U - \psi$. A similar analysis for the rear slip angles gives $\alpha_{3,4} = \alpha_r = (\dot{y} - b\dot{\psi})/U - \psi$. Thus the Newton-Euler equations for the car motion are

$$\begin{aligned} m\ddot{y} &= c_f\alpha_f + c_r\alpha_r + Y^{ext} \\ I_3\ddot{\psi} &= ac_f\alpha_f - bc_r\alpha_r + N^{ext} \end{aligned}$$

where Y^{ext} is the external force in the y-direction and N^{ext} is the external z-moment. The combination of these equations with the equations for the slip angles provides a complete set for the determination of the vehicles motion.

We have to choose a parameterization of orientation to unwind Euler's equation into first order form. If we choose Euler parameters, then we must express $\vec{\omega}$ in terms of the $\dot{e}_0, \dot{\vec{e}}, e_0, \vec{e}$. From Section 2.3.2.4, we know that the orientation matrix M can be written in terms of 3×4 matrices E and G involving the Euler parameters. Thus we could rewrite Euler's equation explicitly in terms of Euler parameters using

$$\vec{\omega} = 2E \begin{pmatrix} \dot{e}_0 \\ \dot{\vec{e}} \end{pmatrix} \quad (2.126)$$

to get a second-order equation in the Euler parameters. This would require the addition of the normalization constraint between the Euler parameters.

Instead of using the Newton-Euler equations directly, we will use the principle of virtual work which is much more adaptable to linked multi-body dynamics. We will separate the integration of the angular accelerations from the integration of the time derivatives of the Euler parameters. We use

$$\begin{pmatrix} \dot{e}_0 \\ \dot{\vec{e}} \end{pmatrix} = (1/2)G^T\vec{\omega}' \quad (2.127)$$

to determine the Euler parameter time derivatives. We combine this Equation 2.127 with the Euler equation to find the body orientation.

2.4.3.2 Virtual Work Equation for a Rigid Body

A virtual work principle is true for a rigid body if we add to the translational work of moving the center-of-mass the work of rotating the body about the center-of-mass by a virtual rotation angle $\delta\pi'$ against any external torques $\vec{n}'^{ext}$. Because we consider only virtual rotations, they are infinitesimal and so are themselves vectors. All virtual displacements used in the virtual work principle must be in the same sense we have used before kinematically admissible. Also, just as in the particle case, we can ignore any internal reaction torques since we will assume they do no work. From d'Alemberts principle, we must also add the inertial terms from the rotational accelerations $\vec{\pi}'^T(J'\dot{\vec{\omega}}' + \vec{\omega}'J'\omega')$. The total additional rotational virtual work term we must add is $\delta\vec{\pi}'^T(J'\dot{\vec{\omega}}' + \vec{\omega}'J'\omega' - \vec{n}'^{ext})$. Notice that we use global coordinates for the center-of-mass motion given by Newton's terms and body fixed coordinates for the Euler terms. To save writing the ', we will abuse the notion (and the reader) and drop the explicit reference to the body-fixed frame for any rotational term in the virtual work equation. This includes the virtual rotation $\vec{\pi}'$. It will be understood that these terms are always given in the fixed body frame.

This gives the virtual work equation for a rigid body

$$\delta\vec{Z}^T(M\dot{\vec{Y}} - \vec{F}^{ext}) = 0 \quad (2.128)$$

where the virtual displacement consists of a kinematically admissible virtual translation $\delta\vec{r}$ and virtual rotation $\delta\vec{\pi}$

$$\delta\vec{Z} = \begin{pmatrix} \delta\vec{r} \\ \delta\vec{\pi} \end{pmatrix} \quad (2.129)$$

where again, $\vec{F}^{ext}$ are the *external* forces and torques acting on the body. As in section 2.2.3, we can ignore workless internal constraint forces and torques.

Example of virtual work with constraint condition: Suppose we have a wheel rolling in one dimension without slipping. Let θ be the rotation of the

wheel from the initial vertical axis. Let x be the hub position of the wheel. The condition for roll-without-slip is $dx + r d\theta = 0$, where r is the tire radius. If the tire has mass m and moment of inertia about its polar axis I , then the virtual work equation with Lagrange multiplier due to the constraint is

$$(m\ddot{x} - f^{ext} + \lambda)\delta x + (I\ddot{\theta} - n^{ext} + r\lambda)\delta\theta = 0 \quad (2.130)$$

where the virtual displacements for position and angle are independent since we have introduced the multiplier. Since the variations are independent the equations decouple to become

$$\begin{aligned} m\ddot{x} &= f^{ext} - \lambda \\ I\ddot{\theta} &= n^{ext} + r\lambda \end{aligned} \quad (2.131)$$

If we differentiate the constraint equation twice and substitute into the equations of motion to eliminate $\ddot{\theta}$, we can solve for λ to get

$$\lambda = -\frac{mrn^{ext} + If^{ext}}{mr^2 + I} \quad (2.132)$$

If the tire is accelerating due either an applied external force or torque there is a non-zero multiplier λ which is the frictional reaction force needed to keep the tire rolling without slipping.

A collection of rigid bodies will have for their virtual work equation

$$\sum_i \delta \vec{Z}_i^T (M_i \vec{X}_i - \vec{F}_i^{ext}) = 0 \quad (2.133)$$

where i refers to the body number and

$$\delta \vec{Z}_i = \begin{pmatrix} \delta \vec{r}_i \\ \delta \vec{\pi}_i \end{pmatrix} \quad (2.134)$$

As with collections of particles, if constraints exist between the rigid bodies, the virtual displacements $\vec{Z}_i$ are not independent, and so the individual body Newton-Euler equations do not decouple. We must, as with the particle case, introduce Lagrange multipliers into the equation in order to assume the $\delta \vec{Z}_i$ are independent.

2.4.3.3 Recipe for Multi-body Equations of Motion

Following ([37]), we give a recipe for multi-body equations of motion:

1. Define the generalized coordinates $\vec{q}$ and determine whether they are dependent or independent.
2. Write each body's Cartesian c.g. position $\vec{r}_i = \vec{r}_i(\vec{q})$ and orientation matrix $A_i = A_i(\vec{q})$ as a function of the generalized coordinates.
3. Write the global body Cartesian velocities $\dot{\vec{r}}_i = \vec{r}_{i,\vec{q}}\dot{\vec{q}}$ and body local angular velocity $\vec{\omega}'_i = B_i(\vec{q})\dot{\vec{q}}$ using vector angular velocity relations or $\vec{\omega} = A^T \dot{A}$.
4. Write virtual displacements and rotations as $\delta\vec{r}_i = \vec{r}_{i,\vec{q}}\delta\vec{q}$ and $\delta\vec{\omega}'_i = B_i(\vec{q})\delta\vec{q}$.
5. Write the global body Cartesian accelerations $\ddot{\vec{r}}_i = \vec{r}_{i,\vec{q}}\ddot{\vec{q}}$ and body local angular velocity $\ddot{\vec{\omega}}'_i = B_i\ddot{\vec{q}} + \dot{B}_i\dot{\vec{q}}$ using vector angular velocity relations or $\ddot{\omega} = A^T \ddot{A}$.
6. Derive applied forces $\vec{f}_i^{ext}$ and torques $\vec{n}_i^{ext}$.
7. Calculate masses m_i and local frame inertia matrices J'_i .
8. Substitute steps 3-5 into the virtual work equation

$$\sum_{i=1}^{nb} \{ \delta\vec{r}_i^T [m_i \ddot{\vec{r}}_i - \vec{f}_i^{ext}] + \delta\pi_i'^T [J'_i \ddot{\vec{\omega}}'_i + \dot{\vec{\omega}}'_i J'_i \dot{\vec{\omega}}'_i - \vec{n}_i^{ext}] \} = 0$$

and rewrite as $\delta\vec{q}^T [\mathbf{M}(\vec{q})\ddot{\vec{q}} - Q^{ext}(\vec{q}, \dot{\vec{q}}) - R^{ext}(\dot{\vec{q}}, \ddot{\vec{q}})] = 0$. Here we have inserted the “”’s, denoting the body-fixed frame, back into the equation for clarity.

9. If the $\vec{q}$ are independent, $\delta\vec{q}$ is arbitrary and $\mathbf{M}(\vec{q})\ddot{\vec{q}} = Q^{ext}(\vec{q}, \dot{\vec{q}}) + R^{ext}(\dot{\vec{q}}, \ddot{\vec{q}})$.
10. If the $\vec{q}$ are dependent, derive the constraint equation $\Phi(\vec{q}) = 0$, the Jacobian $\Phi_{\vec{q}}$, and the constraint acceleration equation $\Phi(\vec{q})\ddot{\vec{q}} = \gamma = -(\Phi_{\vec{q}}\dot{\vec{q}})\dot{\vec{q}}$.

11. The constrained equations of motion are

$$\begin{pmatrix} \mathbf{M}(q) & \Phi_{\bar{q}}^T(q) \\ \Phi_{\bar{q}}(q) & 0 \end{pmatrix} \begin{pmatrix} \ddot{\bar{q}} \\ \ddot{\lambda} \end{pmatrix} = \begin{pmatrix} Q^{ext}(q, \dot{q}) + R(\dot{q}, \bar{q}) \\ \gamma(q, \dot{q}) \end{pmatrix} \quad (2.135)$$

The solution of these equations is non-trivial because of the existence of the holonomic constraint $\Phi = 0$ which must also be satisfied. These equations are not *differential* equations, but *differential-algebraic* equations. In addition, initial conditions must be chosen to be consistent with the constraint equations $\Phi = 0$ and $\dot{\Phi} = 0$.

We will see in the next sections how to implement these equations so that the algorithm can be efficiently used on a parallel computer. The trick is to decompose the vehicle kinematics in such a way that a recursive elimination of the outboard relative joint generalized coordinates can be accomplished. This requires the elimination of all closed loops in the kinematic connections and the substitution of addition constraints to the equations of motion.



Chapter 3

Real-Time Dynamics Algorithms

3.1 Previous Dynamics Modeling Techniques

Computational dynamics is a current very active area of research [34]. Previous work in obtaining equations of motion for a complex system can be generally divided into two categories depending on choice of system coordinates [35]: In the first, no attempt to choose a minimal set of coordinates is made, opting instead for a maximal set of Cartesian coordinates for each body [54]; in the second class, a minimal, or at least small, set of relative coordinates is used recursively to identify a body's position [38]. The first method has the advantages of generality and relatively simple equations - and the disadvantage of rarely being optimal for a given mechanism or vehicle. Commercial codes, such as DADS [68] or ADAMS [18], generate Cartesian coordinates for each body to which a maximal set of constraint conditions must be applied. This is similar to finite element methods in structural mechanics where a very large, but sparse, equation set results. In the second method, a small set of relative joint coordinates is used, and fewer constraints must be applied, but the equations are often extremely complex. A minimal coordinate description is very effective in describing spacecraft and robots, since these systems naturally

form tree graphs. But a problem occurs when closed loops appear in the system graph, since the relative joint coordinates are then subject to additional constraints from the closing of the loop, and so are not independent. Wittenburg [70] considered the idea of cutting the closed loop to form a tree graph, and adding Lagrange multipliers to the equations of motion to satisfy the now missing constraints. Once the system graph is a tree, recursive methods developed by Hooker and Margulies [38] for spacecraft, and Armstrong [4] and Luh, *et al.* [46], for robotics, can be used to recover body accelerations. Bae [5], using these ideas in his thesis, derived an algorithm combining cut joints with recursion. In addition, he discussed issues relating to the natural parallelization of the resulting algorithm. Essentially, we have implemented Bae's work, adding additional techniques for stable integration and pipeline delay reduction to the parallel recursive algorithm.

3.2 Derivation of Recursive Dynamics Equations for Vehicle Model

3.2.1 Rigid Body Kinematics and Euler Parameters

For convenience, we summarize material from Chapter 2. As we have seen, a free rigid body in space has six independent degrees of freedom: Three translational for its center of mass (CM) and three rotational for orientation about its CM. Traditionally, Euler angles are used to describe a body's orientation. But such a parameter choice has singular configurations where its inverse transformation fails to exist. To avoid this potential difficulty, we use four Euler parameters e_0, e_1, e_2, e_3 , describing the angular orientation of the body fixed coordinate frame with respect to the global coordinate frame, satisfying a single normalization condition $e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$. The Euler parameters (which are unit Quaternions) are related to space rotations in the following way: If the rotation has unit axis vector $\vec{\eta}$ and angle θ about the axis, then $e_0 = \cos(\theta/2)$ and $\vec{e} = (e_1, e_2, e_3)^T = \vec{\eta} \sin(\theta/2)$. Euler parameters describe the unit 3-sphere in Euclidean 4-space, and

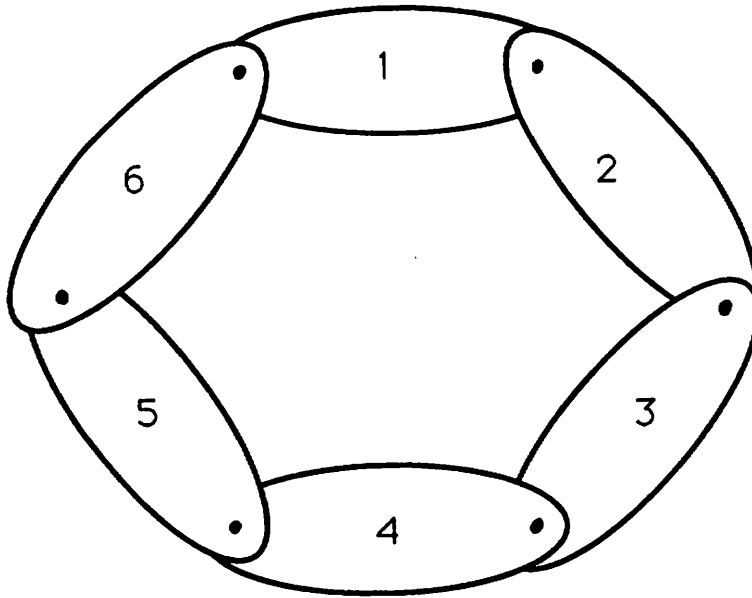


Figure 3.1: Collection of rigid linked together to form a close loop.

so clearly have no singular configuration. For this reason, they are a convenient choice of parameterization when arbitrary vehicle orientations are possible. We combine the Euler parameters as a four component vector $\vec{p} = (e_0, e_1, e_2, e_3)^T$, but we must remember that rotations themselves do not commute and so are not vectors. Simple *algebraic* formulas allow the expression of direction cosine matrices, virtual rotations, and angular velocities in terms of Euler parameters.

3.2.2 Kinematic Relations - The Vehicle Graph

Figure 3.1 shows a multi-body system of rigid bodies linked together by revolute joints which allow only relative rotation about the joint axis between bodies. The bodies are not independent in their motion, but are constrained to move so that the holonomic relations between the coordinates of each linked body are satisfied.

We can abstract this picture and describe the same multi-body system in terms of a *graph*, where the vertices (nodes) represent rigid bodies and the edges joints. This is shown in Figure 3.2. A *directed* graph is a graph where an ordering between vertices is given, so each

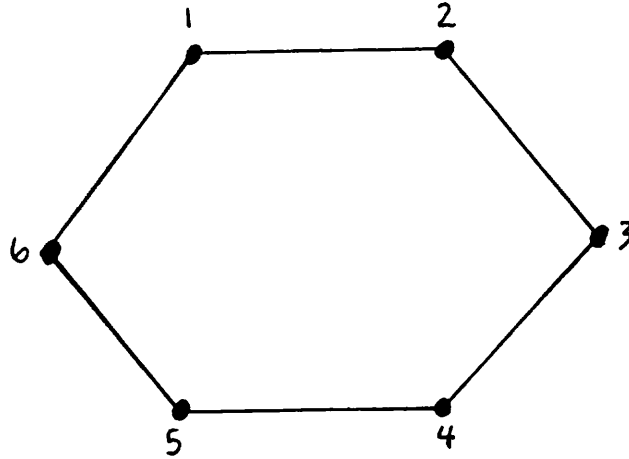


Figure 3.2: Graph associated to multi-body system.

edge has a direction from the ordering of its endpoints. It is possible to be very abstract and prove many theorems about graphs ([61]) useful for automatic generation of code based upon the graph structure of the multi-body system. Here we will be naive and not worry about graph-theoretic formalities. Additional constraint conditions between bodies not from joints are indicated by a dotted line. Figure 3.6 in the next chapter shows the graph for the linked rigid bodies making up a modern sports sedan. The chassis is the *base body* or first body in the directed graph. The constraints imposed by the steering linkage are shown as dotted lines. The linkages themselves are not treated as distinct bodies, but as distance constraints. It is clear that any set of linked rigid bodies can be described in this way, since we make no assumption that the graph have any embedding properties in the plane.

We can now order the bodies by numbering from some starting base body, and derive the kinematic relations between any two linked bodies. Suppose we consider the i^{th} and j^{th} bodies linked together as shown in Figure 3.3. Here we have indicated several coordinate systems: The global frame; body i centroidal frame; body j centroidal

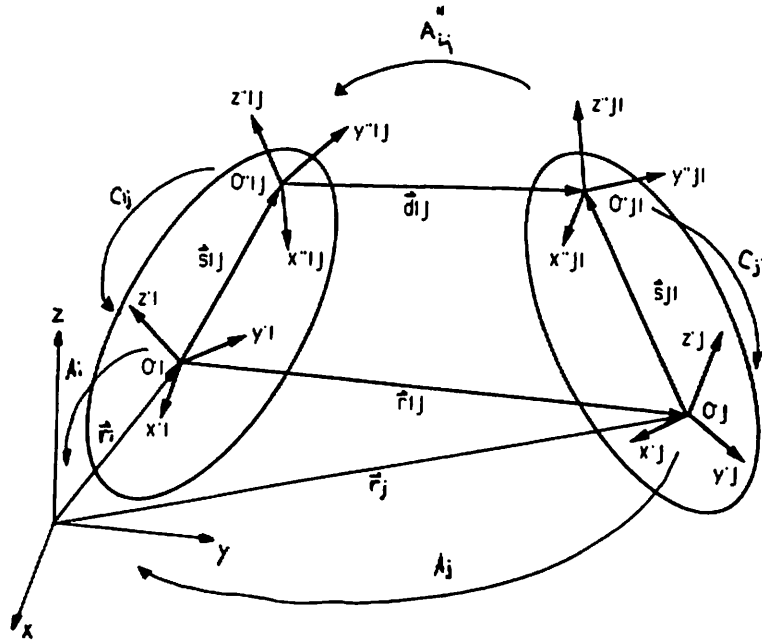


Figure 3.3: Adjacent linked bodies.

frame; the joint frame ij between body i and j ; the joint frame ji between body j and i . $\vec{d}_{i,j}$ is the joint vector between the ij and ji joint frames. The various orientation matrices $A_i, A_j, A_{i,j}'', C_{i,j}, C_{j,i}$ relating the frames are indicated. The global centroidal coordinates of body i are $\vec{r}_i$ and the joint attachment vectors in the local body frame are $\vec{s}_{i,j}$. The A_i are used to transform local vectors like the $\vec{s}_{i,j}$ to global coordinates $\vec{s}_{i,j} = A_i \vec{s}_{i,j}$.

We assume that the joint between the two bodies can be fully described by a set of joint coordinates $\vec{q}_{i,j}$. For example, in a revolute joint, the angle about the joint axis vector; for a spherical joint, the Euler angles or parameters between the ij and ji joint coordinate frames; for a translational joint, the length extension along the translational axis definition vector.

The following relations are easy to deduce:

$$A_j = A_i C_{i,j} A_{i,j}'' C_{j,i}^T, \quad (3.1)$$

$$\vec{r}_j = \vec{r}_i + \vec{s}_{i,j} + \vec{d}_{i,j} - \vec{s}_{j,i} \quad (3.2)$$

$$\vec{\omega}_j = \vec{\omega}_i + \vec{\omega}_{i,j} \quad (3.3)$$

$$\dot{\vec{r}}_j = \dot{\vec{r}}_i + \dot{\vec{s}}_{i,j} + \dot{\vec{d}}_{i,j} - \dot{\vec{s}}_{j,i}, \quad (3.4)$$

where the $\vec{\omega}_i$ are the global angular velocities and $\vec{\omega}_{i,j}$ is the global joint angular velocity between the ij and ji joint frames. A linear relation between $\vec{\omega}_{i,j}$ and the relative joint coordinate velocities $\dot{\vec{q}}_{i,j}$ exists. We write it as

$$\vec{\omega}_{i,j} = H_{i,j} \dot{\vec{q}}_{i,j},$$

where $H_{i,j}$ must be calculated for each joint type. Then

$$\vec{\omega}_j = \vec{\omega}_i + H_{i,j} \dot{\vec{q}}_{i,j}.$$

Since $\vec{s}_{i,j}$ is a vector fixed on body i ,

$$\dot{\vec{s}}_{i,j} = \vec{\omega}_i \vec{s}_{i,j}.$$

For body j ,

$$\dot{\vec{s}}_{j,i} = -\vec{s}_{j,i} \vec{\omega}_j.$$

In terms of the relative joint velocities,

$$\dot{\vec{s}}_{j,i} = -\vec{s}_{j,i} (\vec{\omega}_i + H_{i,j} \dot{\vec{q}}_{i,j}).$$

The derivative of the joint displacement vector $d_{i,j}$ is

$$\dot{\vec{d}}_{i,j} = \vec{\omega}_i d_{i,j} + d_{i,j,q_{i,j}} \dot{q}_{i,j} \quad (3.5)$$

where $d_{i,j,q_{i,j}}$ is the Jacobian of the joint vector $d_{i,j}$ with respect to the relative joint coordinates $\vec{q}_{i,j}$. Putting all this together, we can write the relations between the velocities in terms of matrices $B_{i,j,1}$ and $B_{i,j,2}$ as

$$\begin{aligned} \begin{pmatrix} \dot{\vec{r}}_j \\ \vec{\omega}_j \end{pmatrix} &= \begin{pmatrix} I & (\vec{r}_i - \vec{r}_j) \\ 0 & I \end{pmatrix} \begin{pmatrix} \dot{\vec{r}}_i \\ \vec{\omega}_i \end{pmatrix} + \begin{pmatrix} d_{i,j,q_{i,j}} + \vec{s}_{i,j} H_{i,j} \\ H_{i,j} \end{pmatrix} \dot{\vec{q}}_{i,j} \\ &= B_{i,j,1} \begin{pmatrix} \dot{\vec{r}}_i \\ \vec{\omega}_i \end{pmatrix} + B_{i,j,2} \dot{\vec{q}}_{i,j} \end{aligned} \quad (3.6)$$

A similar formula hold for virtual changes in the coordinates $\delta \vec{r}_i$ and $\delta \vec{\pi}_i$

$$\begin{pmatrix} \delta \vec{r}_j \\ \delta \vec{\pi}_j \end{pmatrix} = B_{i,j,1} \begin{pmatrix} \delta \vec{r}_i \\ \delta \vec{\pi}_i \end{pmatrix} + B_{i,j,2} \delta \vec{q}_{i,j} \quad (3.7)$$

The acceleration relations are

$$\begin{pmatrix} \ddot{\vec{r}}_j \\ \ddot{\vec{\omega}}_j \end{pmatrix} = B_{i,j,1} \begin{pmatrix} \ddot{\vec{r}}_i \\ \ddot{\vec{\omega}}_i \end{pmatrix} + B_{i,j,2} \ddot{q}_{i,j} + D_{i,j} \quad (3.8)$$

where $D_{i,j}$ contains the velocity terms

$$D_{i,j} = \dot{B}_{i,j,1} \begin{pmatrix} \ddot{\vec{r}}_i \\ \ddot{\vec{\omega}}_i \end{pmatrix} + \dot{B}_{i,j,2} \dot{q}_{i,j} \quad (3.9)$$

The matrices $B_{i,j,1}$ and

$$\dot{B}_{i,j,1} = \begin{pmatrix} 0 & (\dot{\vec{r}}_i - \dot{\vec{r}}_j) \\ 0 & 0 \end{pmatrix} \quad (3.10)$$

are independent of joint type, while $B_{i,j,2}$ and $\dot{B}_{i,j,2}$ depend on what the joint and its relative coordinates. Reference [5] has an extensive table of joint types and their B_2 and $\dot{B}_2$ matrices.

Example of revolute joint: Figure 3.4 shows two bodies linked by a revolute joint. Here $d_{i,j} = 0$. Choose the joint axis vector $\vec{h}''$ parallel to the $z''_{i,j}$ -axis and let θ be the rotation angle about this axis. Then

$$\vec{\omega}_j = \vec{\omega}_i + H_{i,j} \dot{\theta}_{i,j}$$

where

$$H_{i,j} = \vec{h}_{i,j} \quad (3.11)$$

$$= A_i C_{i,j} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.12)$$

The joint transformation matrix $A''_{i,j}$ is just the usual rotation matrix about the local ij joint frame by the angle θ . Also,

$$B_{i,j,2} = \begin{pmatrix} \tilde{s}_{j,i} \vec{h}_{i,j} \\ \vec{h}_{i,j} \end{pmatrix} \quad (3.13)$$

and

$$\dot{B}_{i,j,2} = \begin{pmatrix} \dot{\tilde{s}}_{j,i} \vec{h}_{i,j} + \tilde{s}_{j,i} \dot{\vec{h}}_{i,j} \\ \dot{\vec{h}}_{i,j} \end{pmatrix} \quad (3.14)$$

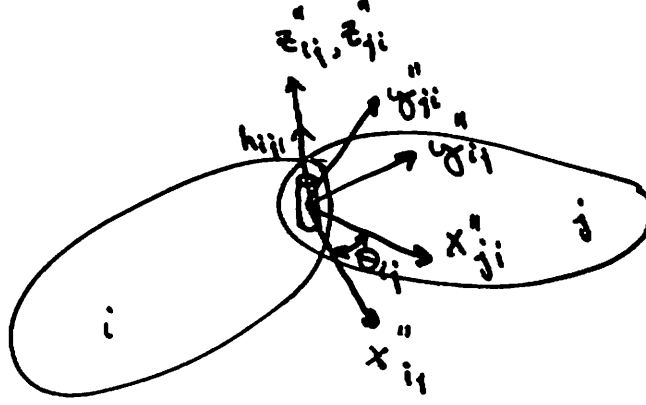


Figure 3.4: Two bodies linked by a revolute joint.

Since

$$\begin{aligned} \dot{\bar{h}}_{i,j} &= \tilde{\omega}_i \bar{h}_{i,j}, \\ \dot{B}_{i,j,2} &= \begin{pmatrix} \bar{h}_{j,i} \tilde{\omega}_j s_{j,i} + \tilde{s}_{j,i} \tilde{\omega}_i \bar{h}_{i,j} \\ \tilde{\omega}_i \bar{h}_{i,j} \end{pmatrix} \end{aligned} \quad (3.15)$$

3.2.3 Virtual Work Equations

The virtual work principle states that for a system of N bodies, the sum of the virtual work (work of a virtual displacement $\delta \vec{q}$ against a force $\vec{Q}$) due to inertial and external forces under virtual kinematically admissible displacements (virtual displacements which satisfy the constraints), added to the the work of the internal forces, must vanish in dynamic equilibrium. For a rigid body we ignore this last term, because the internal forces can do no work. We treat springs and dampers connecting rigid bodies as external force elements.

For a rigid body, we must consider the virtual displacements from both translation and rotation. We follow the notation of [37]. If the

position and orientation are given in the global inertial frame by $\vec{r}_i$ and $\vec{p}_i$, we write the i-th body's global velocity as the vector $\vec{Y}_i$

$$\vec{Y}_i = \begin{pmatrix} \dot{\vec{r}}_i \\ \dot{\vec{\omega}}_i \end{pmatrix}, \quad (3.16)$$

where its rotational velocity is $\dot{\vec{\omega}}_i$; the body's acceleration and virtual displacement-rotation are

$$\dot{\vec{Y}}_i = \begin{pmatrix} \ddot{\vec{r}}_i \\ \ddot{\vec{\omega}}_i \end{pmatrix}, \quad (3.17)$$

$$\delta \vec{Z}_i = \begin{pmatrix} \delta \vec{r}_i \\ \delta \vec{\pi}_i \end{pmatrix}, \quad (3.18)$$

where the *virtual rotation* $\delta \vec{\pi}$ is a true 3-vector. If the system of bodies must satisfy a set of constraint equations $\vec{\Phi}(\vec{r}_i, \vec{p}_i) = 0$, then for every set of *kinematically admissible* virtual displacements $\delta \vec{Z}_i$ satisfying orthogonality with the rows of the constraint Jacobian $\vec{\Phi}_{\vec{Z}_i}$, where this Jacobian notation represents a matrix of partial derivatives of each component of $\vec{\Phi}$ with respect to each coordinate, expressing the linear variation of the constraints in terms of the virtual displacements

$$\sum_{i=1}^N \vec{\Phi}_{\vec{Z}_i} \delta \vec{Z}_i = 0, \quad (3.19)$$

the virtual work must vanish:

$$\sum_{i=1}^N \delta \vec{Z}_i^T (M_i \dot{\vec{Y}}_i - \vec{Q}_i) = 0, \quad (3.20)$$

with the symmetric and positive-definite generalized mass matrix M_i and force vector $\vec{Q}_i$ given as

$$M_i = \begin{pmatrix} m_i & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 \\ 0 & 0 & m_i & 0 \\ 0 & 0 & 0 & J_i \end{pmatrix} \quad (3.21)$$

and

$$\vec{Q}_i = \begin{pmatrix} \vec{f}_i \\ \vec{n}_i - \vec{\omega}_i \times J_i \vec{\omega} \end{pmatrix} \quad (3.22)$$

contain the body mass m_i , inertia J_i , external forces $\vec{f}_i$, and torques $\vec{n}_i$. Geometrically, kinematic admissibility requires every virtual displacement be in the tangent hyperplane to the configuration manifold.

In our case, the virtual displacements $\delta \vec{Z}_i$ are not independent when subject to the constraint conditions Equation 3.19, and so we cannot assume each of their coefficients must vanish. If we add the virtual work of these constraint forces, by means of Lagrange multipliers, then the virtual displacements can be assumed independent. So we replace Equation 3.20 by

$$\sum_{i=1}^N \delta \vec{Z}_i^T (M_i \dot{\vec{Y}}_i - \vec{Q}_i) + \sum_{(j,k) \text{--cut joint}} \delta \vec{Z}_i^T \vec{\Phi}_{\vec{Z}_i}^{(j,k)T} \vec{\lambda}_{(j,k)} = 0, \quad (3.23)$$

where the $\vec{\lambda}_{(j,k)}$ are the Lagrange multipliers from the cut joint between body j and k .

3.2.4 Recovery of Accelerations from the Recursive Equations

We demonstrate (following Bae's thesis [5], [6], [7], [8]) how the recursive equations for the accelerations are derived by considering the single closed loop graph shown in Figure 3.5. Body 1 is the base body, and we cut the loop between bodies l and $l+1$. Equation 3.23 is then

$$\sum_{i=1, i \neq l, l+1}^N \delta \vec{Z}_i^T (M_i \dot{\vec{Y}}_i - \vec{Q}_i) + \sum_{i=l}^{l+1} \vec{Z}_i^T (M_i \dot{\vec{Y}}_i - \vec{Q}_i + \vec{\Phi}_{\vec{Z}_i}^{(l, l+1)T} \vec{\lambda}_{(l, l+1)}) = 0. \quad (3.24)$$

If we now substitute our kinematic relations relating acceleration and virtual displacement of body l in terms of those of body $l-1$ and the relative joint coordinates $\vec{q}_{(l-1)l}$ between them, we get

$$\sum_{i=1}^{l-1} \delta \vec{Z}_i^T (M_i \dot{\vec{Y}}_i - \vec{Q}_i) +$$

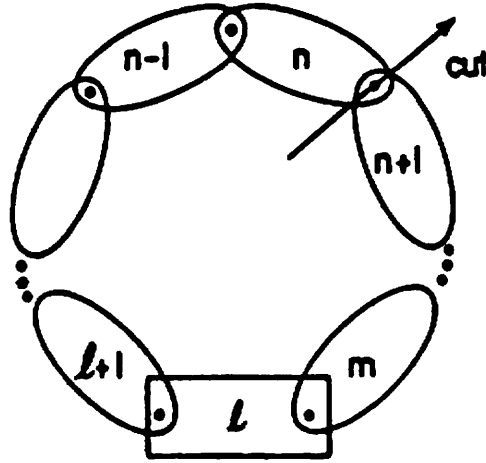


Figure 3.5: Closed graph for derivation of recursive elimination equations.

$$\begin{aligned} & \delta \bar{Z}_i^T (B_1^T (M_l B_1 \dot{\bar{Y}}_{l-1} + M_l B_2 \ddot{\bar{q}}_{(l-1)l} + M_l D_{(l-1)l} - \bar{Q}_i) + \bar{\Phi}_{Z_i}^{(l,l+1)T} \bar{\lambda}_{(l,l+1)}) + \quad (3.25) \\ & \delta \bar{q}_{(l-1)l}^T (B_2^T (M_l B_1 \dot{\bar{Y}}_{l-1} + M_l B_2 \ddot{\bar{q}}_{(l-1)l} + M_l D_{(l-1)l} - \bar{Q}_i + \bar{\Phi}_{Z_i}^{(l,l+1)T} \bar{\lambda}_{(l,l+1)}) + \\ & \sum_{i=l+2}^N \delta \bar{Z}_i^T (M_i \dot{\bar{Y}}_i - \bar{Q}_i) + \bar{Z}_{l+1}^T (M_i \dot{\bar{Y}}_{l+1} - \bar{Q}_{l+1} + \bar{\Phi}_{Z_{l+1}}^{(l,l+1)T} \bar{\lambda}_{(l,l+1)}) = 0. \end{aligned}$$

Since the virtual relative joint displacements $\delta \bar{q}_{(l-1)l}$ are arbitrary (after making the cut), its coefficient must vanish and we can solve for the relative joint acceleration

$$\ddot{\bar{q}}_{(l-1)l} = -(B_2^T M_l B_2)^{-1} (B_2^T (M_l B_1 \dot{\bar{Y}}_{l-1} + M_l D_{(l-1)l} - \bar{Q}_i + \bar{\Phi}_{Z_i}^{(l,l+1)T} \bar{\lambda}_{(l,l+1)})) \quad (3.26)$$

If we substitute the relative acceleration back into Equation 3.26, we have eliminated all body l accelerations. Continuing in this manner backwards along each branch gives an equation for the base body acceleration $\dot{\bar{Y}}_1$ and Lagrange multipliers $\bar{\lambda}$

$$\delta \bar{Z}_1^T ((M_1 + K_1^1 + K_1^2) \dot{\bar{Y}}_1 - (Q_1 + L_1^1 + L_1^2) + (\Phi M_1^{1T} + \Phi M_1^{2T}) \bar{\lambda}) = 0, \quad (3.27)$$

where the superscripts 1 and 2 indicate the result of recursive elimination along each branch to the base body for the accumulated mass (K), force (L), and Jacobian (ΦM) matrices.

If the system has nc constraints, Equation 3.27 does not yet determine the base body acceleration since we have only 6 equations and $6 + nc$ unknown accelerations and Lagrange multipliers. To arrive at the extra equations needed, we differentiate the constraint equations twice to get

$$\begin{aligned} \ddot{\Phi}_{(l-1)l} &\equiv \ddot{\Phi}_{z_l} \dot{Y}_l + \ddot{\Phi}_{z_{l+1}} \dot{Y}_{l+1} - \gamma \\ &= 0 \end{aligned} \quad (3.28)$$

where γ includes all other non-acceleration terms. We can now proceed as before to eliminate recursively the intermediate bodies down to the base body, arriving at

$$(\Phi M_1^1 + \Phi M_1^2) \ddot{Y}_1 + (\Phi L_1^1 + \Phi L_1^2) \ddot{\lambda} - RHS_1^1 - RHS_1^2 - \gamma = 0, \quad (3.29)$$

where again we have accumulated Jacobian mass (ΦM), force (ΦL), and mixed term (RHS) matrices along each branch.

We can combine Equations 3.27 and 3.29 as

$$\begin{pmatrix} K_1 & \Phi M_1^T \\ \Phi M_1^T & \Phi L_1 \end{pmatrix} \begin{pmatrix} \ddot{Y}_1 \\ \ddot{\lambda} \end{pmatrix} = \begin{pmatrix} L_1 \\ RHS_1 \end{pmatrix}, \quad (3.30)$$

where we have collected terms from both branches. Equation 3.30 is then used to find the base body acceleration $\ddot{Y}_1$. Once the base body acceleration is known, we recover each outboard body relative joint acceleration using Equation 3.26 and then the Cartesian acceleration using the kinematic formulas.

3.3 Vehicle Dynamics Model

3.3.1 Real-Time Dynamics for Vehicle Simulation

Evans & Sutherland has devised efficient real-time dynamics algorithms applicable to arbitrary vehicle types. A description of how these algorithm work is fairly complex, but can be broken down into a number of areas.

Non-real-time computer simulation of complicated mechanical linkages has been performed for machines, robots, spacecraft, and ground vehicles. True real-time simulation is more difficult and hitherto done only for relatively simple mechanical systems, such as aircraft, when the governing equations can be approximated, as with linearized equations of motion, or with specialized computer hardware. Engineering simulation of modern vehicles, such as high performance sports cars, requires a multi-body non-linear model in order to provide a high fidelity of motion; software flexibility is required to deal with the general nature of vehicle engineering problems. A solution to real-time engineering simulation demands both cost-effective computational capacity and the adaptability of general purpose computers and software. Simulator system integration and introduction into the real-time software of motion-base washout and vehicle subsystem models are facilitated by the use of standard operating systems and programming languages.

Two significant approaches to real-time vehicle dynamics modeling, albeit at very different levels, are those implemented by the Swedish Road and Traffic Research Institute (VTI) [52] and Daimler-Benz [27] [33]. The VTI model considers a simple planar 3 degree-of-freedom car, but carefully considers tire and other forces. Daimler-Benz achieves real-time with a much more sophisticated, but still simplified model. They incorporate an implicit integration scheme with a quasi-Newton method to achieve stability in a dual processor system.

We are primarily interested in the accurate prediction of vehicle dynamics, rather than kinematics or inverse dynamics [39]. The approach we use decomposes the system into a set of rigid bodies (a rigid body requires distances between points within the body to be

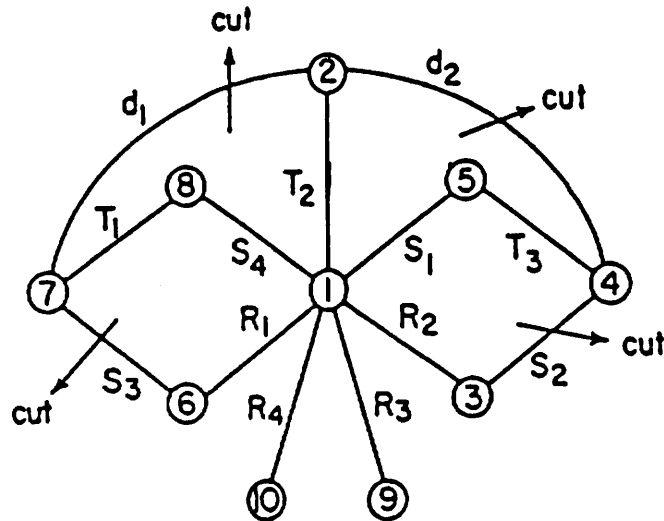


Figure 3.6: Vehicle graph with cut-joints indicated.

fixed) connected by joints (which allow relative motion between the bodies), subject to various forces, moments, and constraints. The resulting linked system can be described by a directed graph [70] (see Figure 3.6), where the bodies form vertices and joints the edges joining the vertices. People and robot arms, usually without closed loops in their connections, form special *tree* graphs contractible to a single base vertex. A tree graph has a natural direction outward from the base vertex. In a tree graph, outboard body (directed out along a branch in the graph) Cartesian positions and velocities can be expressed in terms of inboard body and relative joint coordinates and velocities, recursively providing the coordinates needed for determining motion. We use the term *recursive* to mean either moving inward or outward along the tree graph, using data from previous bodies along the branch to determine the position and motion of the next. General graphs, with embedded closed loops, are required for more complicated systems, such as high-performance sports cars. Our kinematic description of a vehicle allows use of graphs with closed loops. To derive equations of motion for our system, we first cut the closed loops, so that a recursive relative joint coordinate kinematic descrip-

tion can be employed. Next, D'Alembert's virtual work principle [31] is utilized to derive the equations of motion, augmented by reaction forces from the closed loop joint constraints via Lagrange multipliers. The relative joint coordinate formulation, implemented in a recursive parallel algorithm, leads to greater efficiencies for real-time simulation. External forces are computed from the tires, drivetrain, wind resistance, brakes, springs, dampers, stabilizer bars, and suspension stops. Because our vehicle bodies are subject to kinematic constraints, by cutting the closed loops in the graph, we produce a set of coupled differential-algebraic equations (DAE's), in which differential equations of motion and algebraic equations of constraint must be simultaneously solved. At all times, the numerical solution must closely satisfy the imposed kinematic constraints, so some form of constraint stabilization is required. In addition, stability must be preserved even with high frequency force inputs.

In the rest of this section, we describe our recursive vehicle kinematics and equations of motion. Next we treat the numerical integration techniques required for our DAE's, along with the important concept of constraint stabilization. Lastly, we mention computer hardware issues and present the timing results for our algorithm on various serial and parallel machines.

3.3.2 Vehicle Description

A typical modern high-performance sports sedan is shown schematically in Figure 3.7. It consists of a chassis, the two MacPherson front suspension subsystems [15], the steering rack, and the two rear suspension arms. The MacPherson front suspension subsystem is detailed in Figure 3.8. This suspension allows both jounce (spindle vertical motion) and steer (the rotation of the spindle) as independent motions. Steering is done by motion of the tie-rods (which act as distance constraints) connecting the rack and spindle.

The initial step is to break the vehicle into its independent structural components, each of which can be approximated as a rigid body. These will include everything *except* compliant members, such as shock absorbers, bushings, and springs, which have important internal degrees of freedom. These compliant elements can themselves

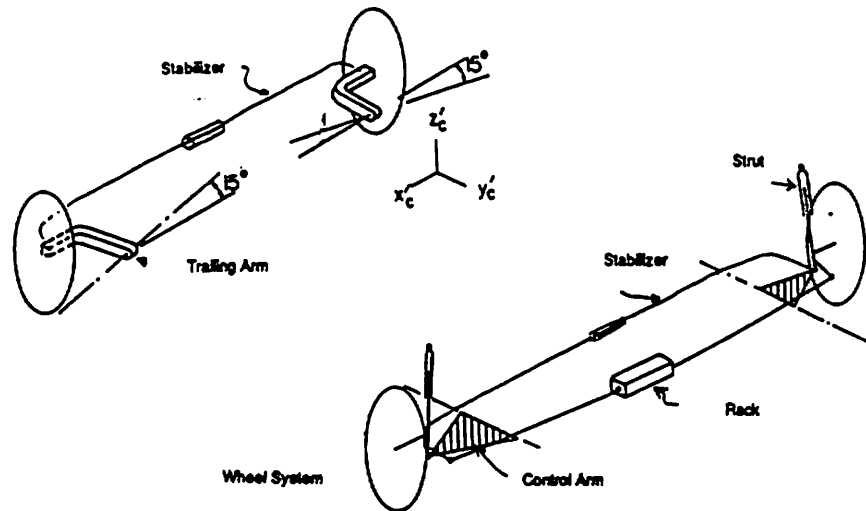


Figure 3.7: Schematic diagram of modern high-performance sports sedan.

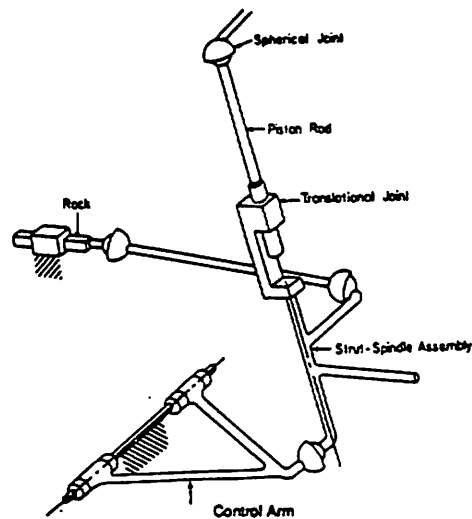


Figure 3.8: Macpherson front suspension.

be modeled using deformation mode analysis, with the coefficient of the mode introduced as another generalized degree-of-freedom. For example, a model might contain 10 independent bodies: Chassis, rack, two Swiss arms, two struts, two spindles, and two rear trailing arms. Shock absorbers and springs are attached to the front and rear suspensions. Torsion bars are included to provide the proper vehicle roll stability.

Each rigid body needs six coordinates to describe its motion: Three for its center of mass and three to describe its angular orientation. Because the rigid bodies are connected to one-another, they are constrained in their motion. For example, the Macpherson front strut assembly has only two motions: The vertical "jounce" and rotational steering. When we account for all the constraints from the joints, we have just *ten* independent degrees of freedom left. The joint constraints effectively convert the dependent individual body coordinates into a smaller set of independent relative joint coordinates. For example, the rear trailing wheel only needs the relative angle between it and the chassis to describe its position. The model takes as its basic variables the chassis Cartesian coordinates and relative joint coordinates. These, along with the associated velocities, are the coordinates that must be propagated through time by the numerical integrator.

3.3.3 Real-time Algorithm

3.3.3.1 Decomposition in terms of a Vehicle Graph

The total vehicle is describable by a *graph* structure: Each rigid body is a node, and the nodes are connected by directed edges which represent the joints. In general, closed loops will exist within the graph. For example, in the front suspension, a closed loop occurs among the chassis, Swiss arm, spindle, and strut. These closed loops will cause problems for the relative coordinates, so they are removed by applying a "cut-joint" constraint in place of an actual joint. This produces a tree-structure graph, without closed loops, but adds additional constraint equations that must ultimately be satisfied by the coordinates.

The corresponding vehicle graph is shown in Figure 3.6. There are four closed loops from the front suspension and steering linkages. If

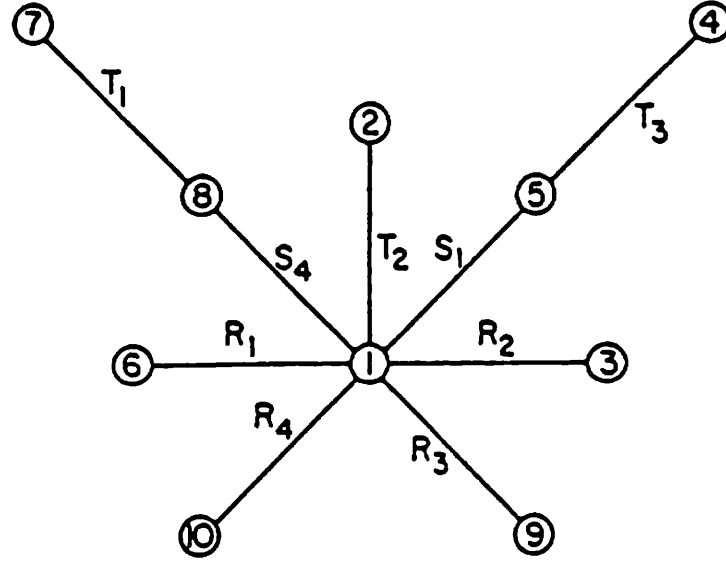
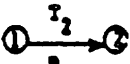
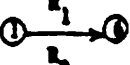
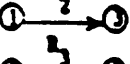
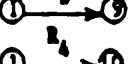
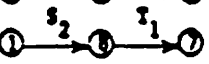
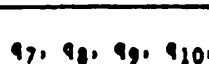


Figure 3.9: Vehicle tree graph with cut loops.

we cut the links between rack and spindle and lower control arm and spindle on both sides, we produce the tree graph shown in Figure 3.9.

3.3.3.2 Vehicle Coordinates

In Figure 3.9, the chassis is the base body, whose coordinates are the three Cartesian degrees of freedom of the center-of-mass in the inertial frame $\bar{r}_1$ and a vector $\bar{\pi}_1$ of four Euler parameters describing its orientation. The lower control arm position is determined by the chassis position and specification of the revolute joint angle (the rear suspension arms have a similar description). The strut's position is given by the chassis and spherical joint orientation (again using Euler parameters). The spindle's position can be found from the strut's position and the translational joint distance. The position of the rack relative to its translational joint with the chassis is given by the driver's steering input. Table 3.1 presents the system's coordinates. Notice that we have not yet "closed" any of the four loops in Figure 3.9 by adding constraints terms to the virtual work equations.

Independent Path	s.c.	joint axis
	q_1	u_1
	q_2	u_2
	q_3	u_3
	q_4	u_4
	q_5	u_5
	$q_6, q_7, q_8, q_9, q_{10}$	u_6, u_7, u_8, u_9
	$q_{11}, q_{12}, q_{13}, q_{14}, q_{15}$	$u_{10}, u_{11}, u_{12}, u_{13}$

$q_7, q_8, q_9, q_{10},$ and $q_{12}, q_{13}, q_{14}, q_{15}$ are relative Euler parameters for spherical joints S1 and S2, respectively.

Table 3.1: Model coordinates.

The four tires are not treated as individual bodies, but are given only a single rotational degree of freedom and inertia. In our model, there are 10 rigid bodies, 4 tires, and 26 body and relative joint coordinates (not all independent) used in the kinematic description of the vehicle tree graph. The constraints from the closed suspension and steering loops must be added to these 26 coordinates. Not treating the tire as a full body is a reasonable approximation, since tire inertial forces can be separately calculated and added as additional external coupling torques to the equations of motion. Compliance in any of the bodies is not considered (cf. remark at the end of the dynamics section).

3.3.3.3 Holonomic Vehicle Constraints

The coordinates used in the vehicle description are subject to holonomic constraints [30] $\bar{\Phi}(\bar{Z}_i) = 0$ (constraints in precisely this functional form, which can be used to eliminate dependent coordinates) from the closed loops in the graph. The MacPherson front suspension requires the spindle to meet the lower control arm at a spherical joint as shown in Figure 3.8. Counting both the right and left

sides, this gives 6 independent constraints. The rack-spindle distance give 2 more independent constraints, while the normalization of the three sets of Euler parameters provides 3 more. The driver steering commands produce 1 more constraint on rack motion. Since each of these constraints is holonomic, and they are all independent, we have $26 - (6 + 2 + 3 + 1) = 14$ independent degrees of freedom in our vehicle (exclusive of the drivetrain and other subsystems).

Non-holonomic constraints (non-integrable relations involving the coordinates) are encountered in our model only in the kinematics of tire roll-without-slip on a surface. The resulting tire constraint forces are treated as external forces on the spindle and tire.

3.3.3.4 Recovery of Outboard Positions, Velocities, Accelerations

We work with a set of 26 dependent coordinates describing the tree graph structure and 12 constraint equations from the cut joints. The velocity and acceleration formulas involve joint matrices B_1 and B_2 giving the linear relation between outboard body velocity and inboard body and relative joint velocity. We write the Cartesian velocity of outboard body $i+1$ as $\dot{\tilde{Z}}_{i+1} = B_1 \dot{\tilde{Z}}_i + B_2 \dot{\tilde{q}}_{(i+1)i}$ where for body i , its global Cartesian coordinates are $\tilde{Z}_i = [\tilde{r}_i^T \tilde{\pi}_i^T]^T$. The corresponding acceleration formula is $\ddot{\tilde{Z}}_{i+1} = B_1 \ddot{\tilde{Z}}_i + B_2 \ddot{\tilde{q}}_{(i+1)i} + \bar{D}_{(i+1)i}$, where the *velocity coupling* terms $\bar{D}_{(i+1)i}$ contain the coordinate velocities and derivatives of the joint matrices B_1 and B_2 . The B_i and D matrices must be individually constructed for each specific joint type.

These kinematic formulas are used to express the global Cartesian position, velocity, and acceleration of an out-board body in terms of an inboard body and relative joint position, velocity, and acceleration. All body coordinates, velocities, and accelerations are determined recursively moving outward in the graph from the initial base body.

3.3.3.5 Recovery of Coordinate Accelerations

We generate our dynamical equations using D'Alembert's principle of virtual work. This is done in a way that generates maximum parallelism in the algorithm. By using relative joint coordinates, rather

than Cartesian coordinates, the unknown joint constraint forces are automatically eliminated from our equations. Using virtual work has the advantage that alternative variational methods require the additional step of forming either Lagrangians or Hamiltonians, and so are cumbersome for complex systems. Added difficulties with variational formulations occur when non-conservative forces (such as friction) are present. With virtual work, we can consider both conservative and non-conservative forces in a straightforward manner, while the cut joint constraints are easily incorporated using Lagrange multipliers.

The result of the equations of motion is a parallel algorithm that allows the inward recursive elimination of the relative joint accelerations, and accumulation of terms into a single matrix equation for the base body accelerations and Lagrange multipliers. The remaining coordinate accelerations are then recursively recovered by moving outward in the graph. The organization of the algorithm closely follows the system tree graph, and so exhibits the inherent parallelism of each independent branch. Accelerations are numerically integrated to determine the new positions and velocities at the next frame time. The recursive algorithm is fully non-linear in that velocity coupling terms $\bar{D}$ are not assumed negligible. The assumption of slow rates of angular change, so coupling from the Euler terms $\bar{\omega} \times J\bar{\omega}$ is small, fails during high speed maneuvering of vehicles. The inclusion of these non-linear terms is critical for simulating the effects of anti-lock braking systems (ABS) [41] and four-wheel steering controls.

3.3.3.6 External Forces

We include various external forces such as tire forces, wind resistance, engine and brake torque, springs, dampers, stabilization bars, and internal reactions in the model. The most critical, and difficult, for accurate road vehicle simulation are the tire forces [20]. The major components of tire forces and moments are shown in Figure 3.10. Because of the complexity of interaction, the forces and moments are taken from measured data as a function of normal force, slip angle, and camber angle. Attention is paid to how the tire forces change during conversion from a no-slip to a slip condition in order to minimize longitudinal force discontinuities.

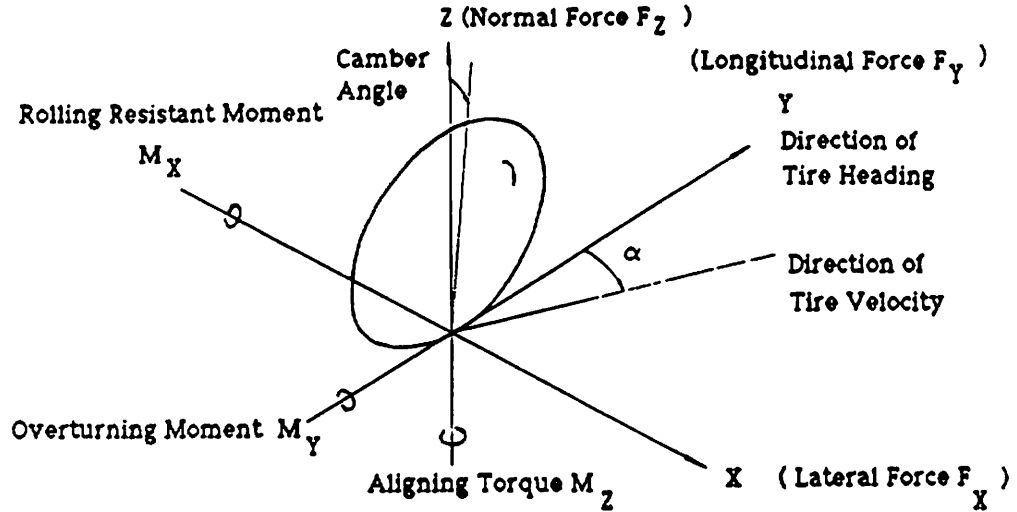


Figure 3.10: Tire forces and moments.

Complex drivetrain models for both manual and automatic transmissions have been developed. They are based on combining drivetrain components such as engine, clutch, transmission, torque converter, and differential together with proper feedback [50]. Each separate component is modeled using table lookups of data provided for individual designs. Our current drivetrain models have a single degree of rotational freedom which is independently integrated to find rotational velocity and applied wheel torques.

3.3.3.7 Integration of Accelerations

We can symbolically write the result of the recursive algorithm determining the accelerations as recovering the right-hand side $\vec{f}(t, Y)$ of the first-order equation

$$\dot{\vec{Y}} = \vec{f}(t, Y), \quad (3.31)$$

where $\vec{Y}$ is the state vector of 26 chassis and joint coordinates and their velocities. This equation includes both the dynamics equations and the differential form of the algebraic cut joint constraints. The resulting DAE's are stiff and awkward to solve [28]. Equation 3.31

can be solved using standard techniques from numerical analysis to advance $\bar{Y}(t)$ to $\bar{Y}(t + \Delta t)$, if combined with constraint stabilization. Because of real-time limitations, methods that require only a single updated evaluation of $\bar{f}$ for each time step can currently be used. This limits us to conditionally-stable explicit methods such as Adams-Bashforth and Adams-Moulton Predictor-Corrector schemes [43] without a final update. Another option is to use low-order Runge-Kutta methods. These have adequate accuracy, with enlarged stability regions. If explicit methods are used, great care must be taken in how the resulting equations are partitioned and solved to avoid instabilities resulting from high frequency force inputs. To improve total system bandwidth, after the system has been properly partitioned into slow (chassis) and fast (suspension) components, a hybrid integration scheme where the low frequency slow components are integrated with an explicit method and the high frequency fast components with an implicit method is advantageous.

Guidance in partitioning the components can be gained by the use of an eigenvalue analysis of the Jacobian matrix for the differential equation system. This linear analysis, though only approximate, will help reveal the potentially troublesome high-frequency oscillatory modes (those with large imaginary parts) and stiff, fast decaying modes (those with large negative real parts). When combined with a stability region analysis of the ODE method, an informed guess as to stable step size can be made. Since it is very difficult to calculate the system Jacobian analytically, a difference approximation about a reference state of interest is used. The zero eigenvalue, rigid body modes must be first removed from matrix before EISPACK or other standard eigenvalue software is used.

3.3.3.8 Constraint Stabilization

The numerical solution to Equation 3.31 must satisfy all cut joint constraints. These additional algebraic equations (forcing us to solve the differential equations on the curved configuration manifold, rather than in flat Euclidean space) require special techniques in correcting the solution produced by the integration method, while retaining accuracy and stability. For short times, since the differential form of the

constraints are incorporated into Equation 3.31, the solution will approximately satisfy the cut joint constraints. For longer times, either Baumgarte's method [9] of modifying the equations to make the configuration manifold (the manifold determined by the constraints on which the system trajectories move) a stable attractor (points close to the configuration manifold move nearer in time along their trajectories), or a coordinate partitioning method [68] of updating the dependent coordinates can be used. Baumgarte's method essentially introduces a feedback loop to control error growth, while the partitioning method uses Newton-Raphson iteration to make the dependent coordinates satisfy all constraint conditions. Either of these ensures that accumulated numerical error will not cause the solution to drift too far off the configuration manifold.

3.3.4 Organization of Parallel Recursive Algorithm

The efficiencies required by real-time demand a careful decomposition of the algorithm data flow and attention to mapping the algorithm to a particular computer architecture. You can't fit a shoe that doesn't fit; you can't expect real-time performance with algorithms and architectures that don't match.

3.3.4.1 Algorithm Flow

Figure 3.11 shows an implementation of the recursive algorithm without constraint stabilization. At the start of each time step, Cartesian positions and velocities of each body are recovered from the relative joint positions and velocities moving outward in the graph from the chassis base body. Each separate branch is computed concurrently with the others. The driver steering, brake and accelerator pedal position and other inputs are obtained from sensors and used to determine rack position and velocity, brake, and drivetrain parameters. External forces and moments are next calculated; tire, brake, spring, damper, stabilizer bar, aerodynamic, gravitational, and drivetrain are independently determined by special subsystem models. Next, the accelerations are recursively eliminated moving inward in the graph toward the base body. Again, these eliminations are performed con-

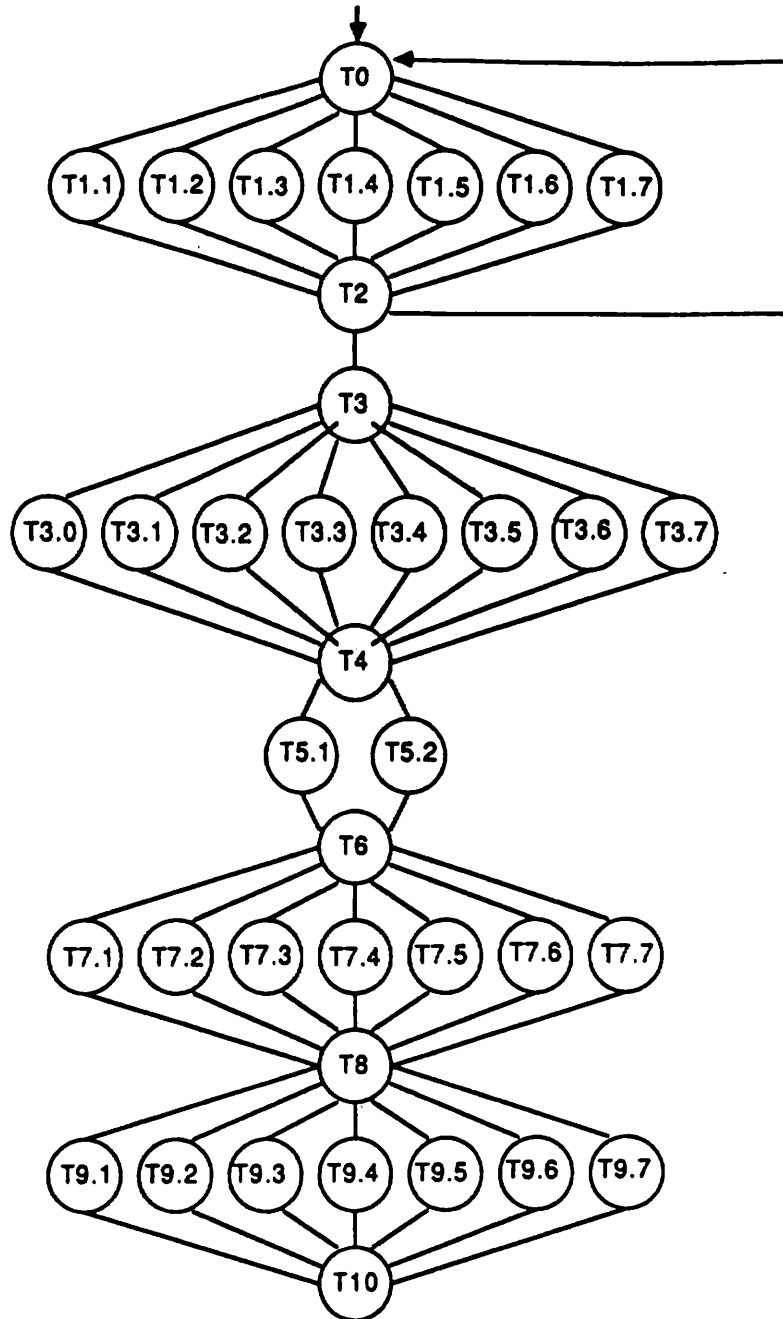


Figure 3.11: Organization of parallel tasking in recursive algorithm.

currently. When the base body Equation 3.30 has been assembled, it is solved for the chassis Cartesian accelerations using Gaussian elimination, and recovery of the outboard body Cartesian and relative joint accelerations is made moving outward in the graph. When this is completed, the right hand side of Equation 3.31 has been calculated and the integration scheme advances the solution to the next time frame.

3.3.4.2 Computer Architectures and Software

The requirement of implementing a parallel algorithm with a small number (4-8) of concurrent tasks leads to considering a single bus or cross-bar architecture with a shared global memory. Care must be taken to avoid potential bus contention from simultaneous memory access or message passing. Modern optimizing FORTRAN and C compilers running under POSIX-type operating systems provide software flexibility. Specialized simulation computers with non-standard languages suffer from insufficient adaptability, limited transportability, and narrow growth potential. Upgrading of a simulator computer system is simplified by writing software in standard languages.

A solution to these problems is to choose a standard vector-concurrent mini-supercomputer, like the Alliant FX/8, running a UNIX operating system. Figure 3.12 shows the architecture of the Alliant FX/8 ([2]).

In the Alliant FX/8, up to 8 compute elements (CE) are connected by a 376MB/sec cross-bar to two large computational processor caches (CPC). Each CPC is a two-way interleaved 256KB cache for up to 4 CEs. Cache coherency is maintained by a write-back cache protocol¹ watching the main memory bus. The CPC are connected to a 4-way interleaved 64MB main memory by two independent 188 MB/sec 72 bit (64 bits data and 8 bits SECDED) data buses, a 28 bit address bus, and a concurrency control bus. Each individual CE contains an integer, floating point, and vector register set. The floating point units

¹A write-back cache is one where writes to memory are stored in cache and written to memory only when a rewritten item is removed from cache. In a write-through cache, writes to memory are written concurrently to both the cache and memory.

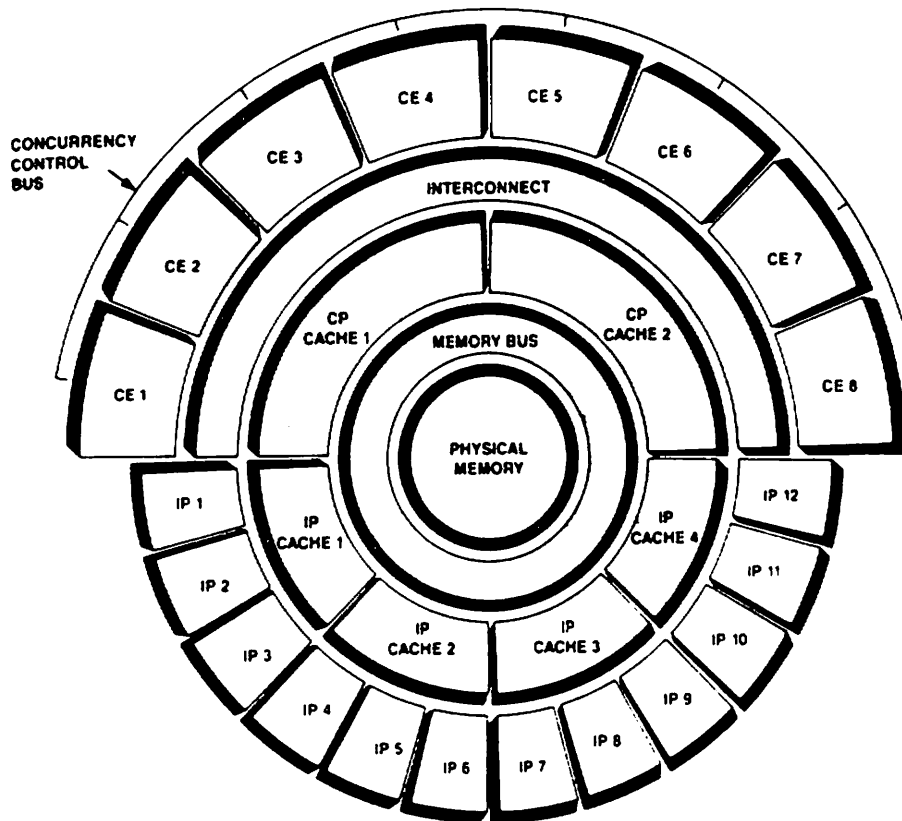


Figure 3.12: Organization of the Alliant FX/8.

are fully segmented, and can perform simultaneous scalar multiply-adds. Special vector operations are available that will chain standard vector calculations using adds and multiplies together. All floating point operations are in 64 bit, and the vector lengths are 32 64-bit vectors (CRAY has a length of 64 64-bit vectors). The peak "macho-flops", with all 8 CEs, are 188 MFLOPS. A more realistic number would be 20-40 MFLOPS sustained.

The operating system and all I/A operations are done with interactive processors (IP). These are essentially Motorola 68020 units with some possibility of doing floating point operations. These IPs access main memory through the IP caches (IPC), each of which share up to 3 IPs. Each IP also has a local memory to hold operating system kernel text pages. Direct memory access (DMA) is available to VME devices through special VME IPs. The design of the Alliant allows computation to occur concurrently with I/A operations. If the majority of the executing process is cache resident, I/A operations have no effect on system performance. This is important for a simulator, since a modest amount of (less than 100 bytes) output must be delivered without slowing the real-time computations.

Concurrency control is done through the existence of concurrency registers on each CE, and global control of the concurrency is done by concurrency instructions communicating via the concurrency bus. Synchronization of parallel execution of loops does not require access to main memory by the use of spin-locks and semaphores, but is done directly in the concurrency registers.

Very good optimizing vector-concurrent FORTRAN and C compilers are available for this class of architecture. They are limited in their detection of parallelism by the inability to fully trace all data dependencies, and so really only automatically produce COVI - concurrent outer, vector inner - loop breaking. If the algorithm does not allow for its description in terms of single or multiple loops, the programmer must provide explicit directions to enable the compiler to parallelize the code. If Execution of multiple concurrent threads is required, a time consuming analysis of the algorithm data flow must be performed, resulting in a decomposition into serial groups of parallel tasks. The compiler can then proceed to automatically insert instructions for the required data coherence and task synchronization. The

RELATIVE TIME TO FINISH ONE ALGORITHM STEP
(ALLIANT FX/8 WITH 8 CE'S = 1 UNIT)

Alliant FX/8	(8 CE's - parallel code)	1
Alliant FX/8	(4 CE's - parallel code)	1.25
Alliant FX/8	(2 CE's - parallel code)	2.0
Alliant FX/8	(1 CE - serial code)	3.2
VAX 8800	(serial code)	4.5
VAX 780	(serial code)	21.5

Table 3.2: Timing results for recursive algorithm.

goal of the algorithm analysis and optimization is to provide as many task groups of up to 8 course-grained (to reduce the need for memory conflicts over shared data) concurrent tasks as possible, with the smallest possible number of single serial tasks separating the concurrent task groups. Such a design will maximize the parallel speed-up, hopefully approaching one linear in the number of processors. Practical algorithms rarely achieve linear speed-up. The recursive algorithm on the Alliant FX/8 achieves a speed-up of 3.2 with 8 CEs (cf. Table 3.2).

3.3.4.3 Algorithm Timing

The recursive algorithm for the vehicle model described in section 3.3.3 has been run on the Alliant FX/8 and VAX-780.

Table 3.2 shows the results of timings taken on these computers, both serial and parallel, using optimized FORTRAN codes for each machine.

3.3.4.4 Remarks

Two remarks:

1. Notice that the way the algorithm proceeds, a correction to tire position occurs. We do not just assume a tire position, but a step is taken to correct it, and improve the reaction force calculations.
2. The algorithm, being organized around a tree structure without closed loops, clearly lends itself to a parallel implementation. The left-right symmetry produces the greatest improvement in speed, with additional marginal improvements among each set of branches on each side.

Chapter 4

Physiology for Simulation

4.1 Effective Simulation

Effective simulation requires that the simulated environment approach that of the real world as closely as possible. The various sensors of the human subject need to be stimulated in such a way that proper correlation between all the sensors is achieved. Consideration of human factors in the design of the visual and motion systems are critical for a believable environment.

Achieving the desired environment requires an understanding of the human perception system and its interaction with the simulator. One needs to understand the physiology of the various sensory organs to be able to coordinate the stimuli given to the subject so that valid results can be obtained. A simulator is of little use if it elicits an improper response to the presented scenario.

First, a review of linear systems theory will be presented as it pertains to modelling the humans sensory systems and motion control. Next, a discussion on the human perception systems will be presented followed by issues regarding the design of an effective simulator motion system.

4.1.1 Linear Systems Theory

In subsequent sections, linear systems theory will be used to describe several different systems. We want to provide a brief review of the

theory and then illustrate the concept with an example.

The first step in describing a system is to derive a set of linear, constant coefficient, ordinary differential equations (ODE). This set of equations describe the dynamics, or behavior, of the system.

The first tool that will be used is the Laplace transform. In this transform we replace derivatives with the Laplace variable, s . The main advantage is that this transform puts the ODE's into the frequency domain in a manner similar to the Fourier transform and transforms them into algebraic equations. These can then be manipulated with the ease of any algebraic equation. All the standard linear system tools, such as frequency response graphs, are available for system analysis and design.

The next step is to define the system transfer function. The transfer function defines the input-output relationship of a linear, constant coefficient system. It is important to note that it is only defined in the Laplace domain. Once the system transfer function is derived, the large toolbox of linear systems analysis tools are available to describe and characterize the original system.

If the reader wants a more in depth review of linear systems we recommend Dorf, [25] or Kuo, [42] for control systems. For digital filters we suggest Oppenheim, [53] and for general linear systems, Strang [66] is a good reference.

Example of the Laplace transform method for the differential equations of a spring, mass, damper system:

The equation of motion for a generic spring, mass, damper system is:

$$M \frac{d^2 y}{dt^2} + c \frac{dy}{dt} + K y = r(t) \quad (4.1)$$

Where M is the mass, c is the damping, K is the spring constant, y is the displacement of the mass, and $r(t)$ is the forcing function.

Rearranging into the more familiar form with the natural frequency, $\omega_n = \sqrt{\frac{K}{M}}$, the damping ratio, $\zeta = \frac{c}{c_c}$, and the critical damping, $c_c = 2M\omega_n$:

$$\frac{d^2 y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = \frac{\omega_n^2}{K} r(t) \quad (4.2)$$

The Laplace transform of 4.2 is:

$$s^2 Y(s) + 2\zeta\omega_n s Y(s) + \omega_n^2 Y(s) = \frac{\omega_n^2}{K} R(s) \quad (4.3)$$

Where $Y(s)$ is now the transformed displacement, s is the Laplace variable, and we have assumed that the initial conditions are zero. The transfer function is found by forming the ratio of the terms multiplying $Y(s)$ and the terms multiplying $R(s)$.

$$\frac{Y(s)}{R(s)} = G(s) = \frac{\omega_n^2}{K(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (4.4)$$

Not all systems can be adequately described by linear systems theory. Even systems with mild non-linear behavior can display behavior that is not a small perturbation of some linearized model.

4.1.2 Human Perception Systems

The human has many sensors to allow us to define and interact with our environment. Proper stimulus of these sensors is the key to creating a realistic environment. A proper understanding of the function and modeling of these systems is essential to their proper stimulus.

The three basic groups of sensors that will be discussed are:

1. The eyes.
2. The vestibular system or middle ear.
3. The proprioceptive sensors.

4.1.2.1 The Eyes

The eyes are perhaps the most complicated system we have for perception. In the simulator, the eyes are used as a primary source of information about the environment. A complete exposition of the eye and its functions is well beyond the scope of this paper. The reader is referred to [12] and [13] for an extensive discussion of the eye as a

detector. Only general issues relating to the eye and its involvement in the simulation will be discussed.

The eyes are sensitive to positions and velocities. Regan in [56] demonstrates that the eyes do not really detect linear accelerations, nor do they differentiate velocity to derive acceleration. The eyes are most sensitive to velocity in the peripheral vision so the computer image generator (CIG) should have a wide field of view, 50 degrees or more, to generate adequate velocity cues. Care must be taken at these viewing angles to avoid over stimulating the peripheral vision if motion is not to be used. These wide angle views tend to induce simulator sickness especially when high scene density is present.

The CIG needs to meet certain requirements. According to Casali, [16], the CIG should have proper priority¹ and anti-aliasing hardware and software. He also cites other factors contributing to problems in visual perception including double imaging or ghosting, and chrominance discontinuities between adjacent CRT's or screens. When the luminance level is too low, it contributes indirectly to perceived distortion because the pupil will dilate causing an increase in spherical aberration.

Most image systems today are *biocular*² rather than *binocular*³. The human vision system is inherently binocular. It is through this binocular vision that we perceive depth and distance to objects. A biocular CIG system has inherent problems conveying depth and distance to objects in the display. The eye must be made to focus at the proper distance otherwise velocity and distance cueing become significant problems. Infinity optics can be used with CRT displays to help alleviate this problem, but screen displays have to rely on the perspective given to the displayed objects for depth cueing.

4.1.2.2 The Vestibular System

The Vestibular system is comprised of the semicircular canals and the otoliths. It makes up the non-auditory portion of the ear and is

¹An assignment to a displayed object which determines whether it is displayed in front of or behind other objects.

²In a biocular system, both eyes share a common optical axis.

³In a binocular system, each eye has its own optical axis.

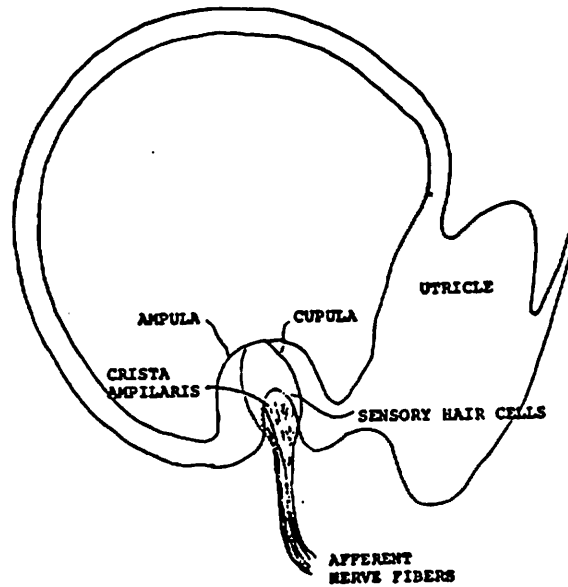


Figure 4.1: The Semicircular Canals

our primary means to measure linear and angular acceleration. The output of these sensors is intimately linked to and correlated with the motion our eyes detect. The correlation between vestibular detected motion and visual perceived motion must be maintained in the simulator environment for effective simulation.

The semi circular canals are sensitive to angular acceleration and the otoliths detect linear acceleration. There is some sensitivity to linear acceleration in the semicircular canals as well as the otoliths to angular acceleration. However, the brain is able to resolve this coupling so that there is little problem distinguishing linear from angular acceleration.

Semicircular Canals A typical semicircular canal is shown in Figure 4.1, taken from Riedel, [60]. There are three semicircular canals in each ear arranged in approximately orthogonal axes. Each canal is filled with a fluid called endolymph which is similar to water in its physical properties. At the base of each canal is a chamber called the ampulla. Inside the ampulla is a gelatinous mass called the cupula which completely blocks the canal and is attached to nerve fibers

through hair like sensory cells. When the head is subjected to an angular acceleration, the endolymph displaces relative to the canal walls and in turn displaces the cupula. This in turn changes the afferent firing rate of the sensory cells as they bend. The sensory cells in any one canal have the same polarization and so all will have a comparable change in firing rate for any given displacement, [57,58,60,12,13].

The usual model for the semicircular canals is a torsional pendulum. The semicircular canals act like integrating accelerometers and integrate angular acceleration to obtain angular rates for normal velocities and durations of head motion. The transfer function for a typical model is, [57,58]:

$$G(s) = \frac{T_L T_a s^2}{(T_L s + 1)(T_S s + 1)(T_a s + 1)} \quad (4.5)$$

The input is the angular rate and the output is the perceived angular rate. The parameters T_L , T_a , and T_S are interpreted as time constants and s is the Laplace variable. Reid, [57,58], uses the following values; 10.2 sec., 30.0 sec., and 0.1 sec. respectively. The frequency response of this transfer function is given in Figure 4.2.

The Otoliths The otoliths are separated into a saccule and utricle with each ear having one of each. A typical saccule or utricle is shown in Figure 4.3, taken from Riedel, [60]. The otolith is a shell filled with calcium carbonate crystals, octonia, surrounded by endolymph. This system is supported on sensory hair cells similar to those in the semi circular canals. When the head is subjected to a linear acceleration, the octonia displaces relative to the endolymph changing the mass distribution. This change causes the sensory hair cells to deflect which, in turn changes the afferent firing rate.

The utricle is located in a plane that is roughly parallel to the horizontal semicircular canal. Its sensory cells are sensitive to motion in all directions in that plane. The saccule is located in a plane approximately perpendicular to the utricle. However, its sensory cells are more sensitive to acceleration perpendicular to the utricular plane, [57,58,60,12,13,11]

The otoliths detect specific force, which is the difference between the induced acceleration and the acceleration of gravity. The typical

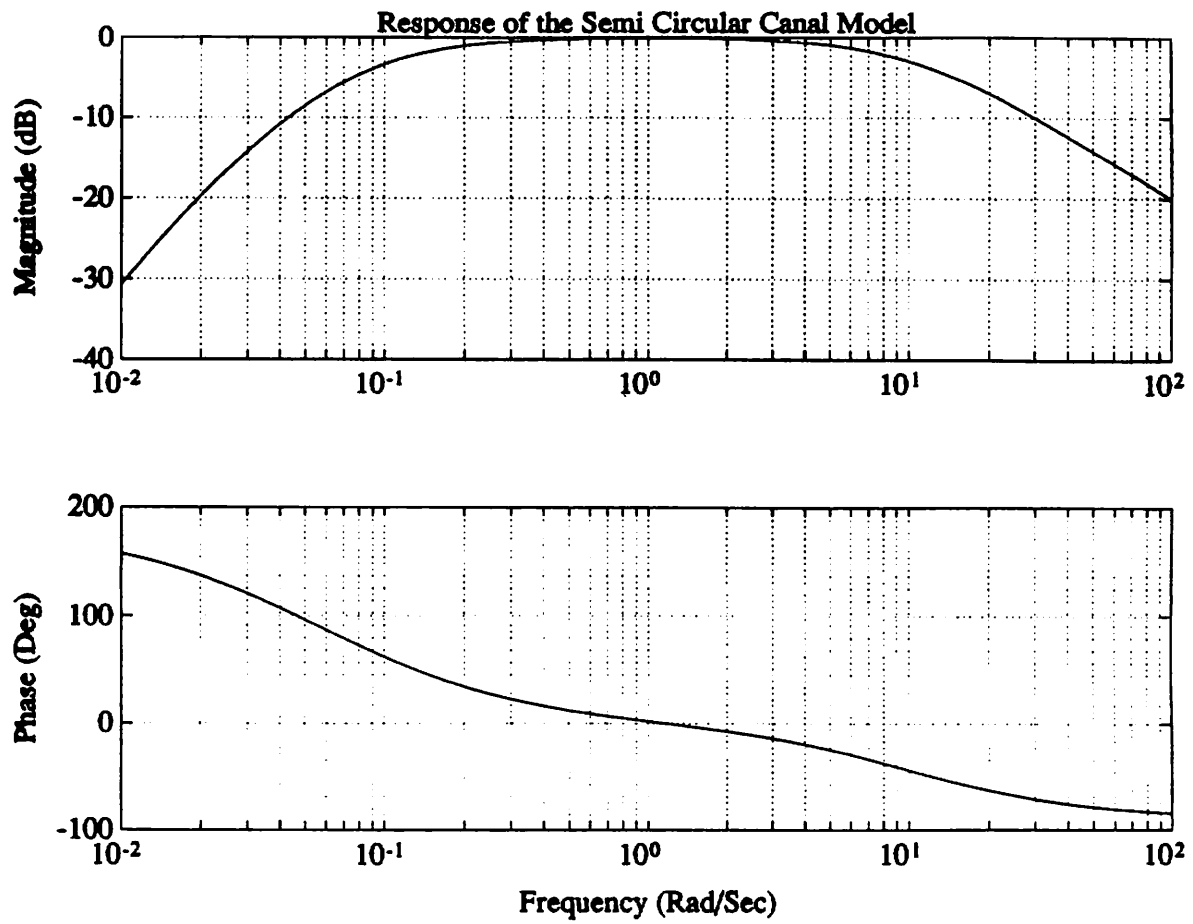


Figure 4.2: Frequency Response of the Semicircular Canal Model

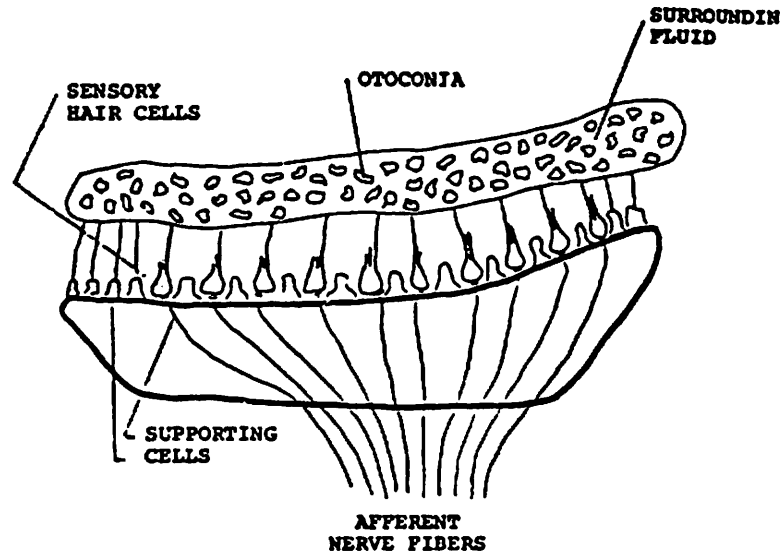


Figure 4.3: The Otolith

model for the otoliths is a mass balanced accelerometer. Conceptually, a mass balanced accelerometer is a device that measures acceleration indirectly by the deflection of a beam with a known mass suspended at its end. When it is accelerated, an inertial force due to the mass causes the cantilever arm to bend. The induced strain of the arm is used as an indication of the magnitude of the acceleration. Following Reid [57,58], the transfer function model of the otolith is:

$$G(s) = \frac{K (\tau_a + 1)}{(\tau_L s + 1)(\tau_S s + 1)} \quad (4.6)$$

The parameters τ_L , τ_a , and τ_S are interpreted as time constants, and s is the Laplace variable. Reid, [57,58] assigns the following values; 5.33 sec., 13.2 sec., and 0.66 sec. respectively. The gain K is given the value of 0.4. The frequency response of the otolith model is given in Figure 4.4.

4.1.2.3 Perception Thresholds

In a simulator, there are certain motions that must be accomplished without the subject being aware. For example, if the simulator is to

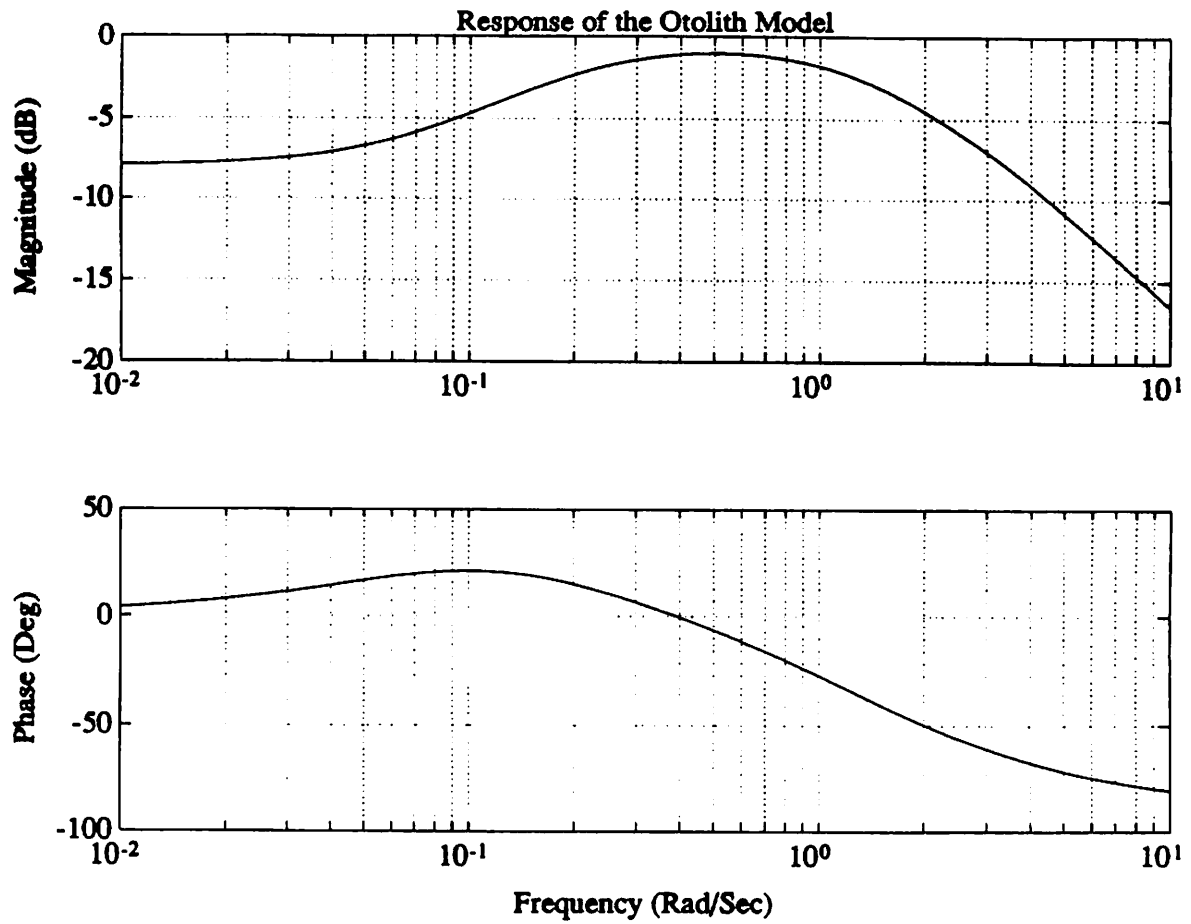


Figure 4.4: Frequency Response of the Otolith Model

be tilted to simulate a sustained linear acceleration, the rotation must be done at rates that are imperceptible to the subject. A major cause of simulator induced sickness is when motion that does not correlate with the visual perception is detected by the vestibular system.

One of the chief problems with assigning a threshold is that there is quite a variation in the laboratory results. Examining various references such as Berthoz [11], the *Handbook of Human Perception and Human Performance* [12], or *Engineering Data Compendium: Human Perception and Performance* [13] will show this difference. Experimental information is determined in very controlled circumstances often with only one motion axis being excited at any one time. The experiments are also often performed in the dark in soundproofed rooms to eliminate any external stimulus. Fortunately, there are other considerations in a simulator which allow us a fair amount of leeway in choosing thresholds. In contrast to the above experiments, the simulator environment is actively stimulating many different sensory inputs. Although masking of motion in one degree of freedom (DOF) is not necessarily masked by motion in another⁴, the high density of information from all the subject's sensors tends to raise all the thresholds.

In practice, common threshold levels for the semicircular canals are in the range of 2.5 to 3.5 degrees/sec and 0.15 to 0.3 meters/sec/sec for the otoliths, [11,12,13,57,58]. In the end, knowing the magnitude of these thresholds is the starting point. The actual values used are best determined by experimentation in a specific simulator given a particular experiment.

4.1.2.4 Proprioceptive and Tactile Sensors

The proprioceptive sensors are those sensors in the muscles, tendons, and joints that respond to stimuli from within the individual. For example in the base of the neck there are sensors that help orient the

⁴In recent experiments, Greig, [32] indicates that there may not be a threshold per se, but rather that motion detection is better represented by a signal-in-noise model and analyzed with signal detection theory. Gaussian receiving operating characteristic (ROC) curves are used to define the detector's performance. The detection of motion becomes a statistical question rather than one with absolute thresholds. The reader is referred to Greig, [32], for the complete study.

head. The proprioceptive sensors are normally considered secondary to the eyes and vestibular system with regard to motion detection.

Tactile sensors respond to skin pressure or touch. Again, these are secondary in importance to the vestibular system and eyes.

In certain circumstances, a jet fighter simulator for example, a gross motion system is not practical. The accelerations are just too large and of too long duration to be simulated with any realism. Rather than not having any sense of acceleration, "g-suits" are often used. These suits can be pressurized in various ways to stimulate the proprioceptive and tactile sensors giving some feeling of acceleration.

4.1.3 Simulator Sickness

Simulator induced sickness (SIS) is the target of much concern. A simulator rapidly loses its utility if the subjects are forever becoming ill during the course of an experiment. Understanding SIS is crucial in designing a simulator that will minimize its occurrence.

SIS shares many common symptoms with motion sickness but is more than just a subset of motion sickness. Shared symptoms include nausea, pallor, fainting, and a change in respiration and heart rate. There are, however, other factors that are exclusively related to SIS. For example, SIS can occur when there is no motion at all. Anyone who has experienced a very wide field of view motion picture can relate to the feelings that this sort of visual stimulation can induce. Visual flashbacks are another symptom exclusive to SIS. These flashbacks are a concern because they are dangerous. They cause disorientation and can occur hours after the simulation has ended. If they happen when the subject is for example driving a car, a serious accident could occur.

The causes of SIS are all related to the fact that the simulated environment is not the same as the real world. Miscues, when vestibular and visual motion do not correlate, are a primary cause of SIS. Delays in motion onset with regard to the visual motion onset are an example of miscueing. The motion of the simulator must be synchronized with the visual perceived motion to avoid miscueing.

Inadequacies in the CIG are suspected as possible causes of SIS as well. Casali in [16] lists such things as flicker, off axis viewing,

parallax, and image smearing as contributing factors to SIS.

Sometimes, it's the high fidelity CIG's that are more prone to inducing SIS. Casali [16] discusses data that demonstrate that a wide field of view CIG with a high scene density is more prone to induce SIS than a CIG with low scene density *when no motion is used*. It is thought that the reason for this is that the high fidelity system stimulates the subject's vision much more than the low fidelity system causing a greater discrepancy between the vestibular system and the eyes. If a high scene density CIG is used, motion should almost always be used to some degree to help avoid SIS.

Subjects with a lot of experience in the vehicle being simulated are more prone to SIS than those without the same expectations. The vestibular system is very adaptable and becomes conditioned to expect a certain excitation in familiar circumstances. In a simulator, this excitation will often be different. This discrepancy in expected and experienced stimulus can result in SIS especially if the exposure time to the simulation is extended.

Masking false motion cues or making up for the lack of motion cues is not always possible as Greig, [32], discusses. There has been some success, however, found by the army in a recent venture in tank operator training. The army's SIMNET⁵ experiment found that intense psychological and aural loading of the subjects in the simulator made up nicely for the lack of motion and low grade CIG's. It's thought that the intensity of the environment occupied the thought processes of the participants to the point that the lack of other cues was not noticed.

It is possible that in other circumstances that a highly intense emotional environment can mask the lack of correlation in cueing and thereby allow more latitude in motion system and CIG design.

In summary, any difference between the real world and the simulator experience is a possible cause to SIS. It is even more apparent when the cues from different sensors do not correlate. The basic phi-

⁵SIMNET is a networked tank training environment. There are several simulators linked into one data base. There is no motion and the CIG's are very basic. SIMNET is used to train tank crews in a giant electronic war game. All the expected intense psychological pressure and aural distraction from tank noise, artillery, and radio interaction that would be found in a real battle is recreated for the simulation.

losophy for avoiding SIS is to recreate all cues as faithfully as possible and most importantly, correlate all the cues as well as possible. False cues need to be masked or eliminated. If motion contrary to the visual motion is required, that motion must be at levels imperceptible to the subject.

4.1.4 Designing an Effective Simulator Motion System

The design considerations of a simulator system are not completely obvious nor are they trivial. A complex interaction exists between the CIG, motion system, the vehicle's computer model, and the subject in the simulator. This interaction must be considered carefully and completely in the design of a system.

There are always trade-offs in any design effort. A simulator motion system can be a very large, heavy, and complex system. Payload masses of up to 7000 Kg or more are conceivable when one considers the mass of a dome, projectors, cab, and other supporting structures. When this mass is supported on a motion system, large moments of inertia can result. These large masses and moments of inertia create problems for sizing the hydraulics and compensation in the servo controller. Careful thought and design has to be used in order to minimize the cost and physical size of the simulator and hydraulic systems.

Another trade-off is related to the motion required for the vehicle being simulated. The platform performance envelope must be matched carefully to the performance envelope of the vehicle being modeled. An aircraft has different predominant motions than a land based vehicle. The maneuvers are different and the control is different. A motion system that is optimal for an aircraft is most certainly sub-optimal for an automobile. A careful evaluation of the performance and characteristics of a vehicle must be done prior to the design a motion system to simulate that vehicle.

Even when the simulator motion system is optimal, it will still not be able to exactly replicate all the motions of the vehicle. The output of the vehicle dynamics needs to be processed so that the motion system is commanded in such a way as to make the best use

of the available performance at the same time giving the proper cues to the subject in the cab. A *washout filter* is used for this purpose. A fundamental piece of the washout is a high pass filter. The high pass function removes the low frequency components of the motion commands that are the most likely to cause performance limits to be exceeded. Motion commands that have a frequency content typically of 2 Hz or lower are those that cause the most problems.

Washout filters deserve special attention because of their key role in the effectiveness of a simulator.

4.1.4.1 Washout Filters

A washout filter is more than just a digital filter. It provides rate and acceleration limiting when needed to keep certain motions below thresholds, as well as providing the means of simulating those motions that are beyond the capabilities of the motion system. Washout filters play a critical role in how effective the simulator's motion system is.

Tilt Coordination The hardest motion to simulate is a sustained acceleration. Eventually, any simulator motion base will run out of displacement capability and sustained accelerations quickly push a simulator to its displacement limits. The most common method used to simulate sustained accelerations is with *tilt coordination*. Sustained lateral and longitudinal accelerations are just not feasible using most standard simulator motion bases. An obvious solution is to tilt the simulator and let gravity make up for the lack of performance. This is the basic principal of tilt coordination. There is one significant difficulty with this, if the simulator is tilted too fast, the subject will be misued and could succumb to SIS.

Tilt coordination has had good success in flight simulators because of the very nature of the maneuvers performed in an aircraft. In general, only large relatively slow response aircraft use motion systems. This slow response allows time for tilt coordination whereas in an automobile, the response is much faster and the maneuvers are more sudden. The onset and termination of accelerations are very fast so there is just not time for proper tilt coordination.

Imagine coming to a sudden stop. The simulator will typically run

out of displacement and tilting the simulator is a natural solution. However, when the brake is released, the acceleration is terminated almost instantaneously. If the simulator remains tilted, the subject still feels that he is stopping. If the tilt coordination is terminated quickly, the subject will perceive the rotation and a miscue will result. Hahn and Kalb in [27] found that tilt coordination was a problem in their experiments in the Daimler Benz Simulator. They suggest that information from the visual data base could be used to anticipate the possibility of sustained lateral and longitudinal accelerations. These data could be used to intelligently make use of the simulator's performance envelope. Normally though, tilt coordination is not useful in a driving simulator.

The correlation of the tilt coordination is achieved through a low pass filter. The low frequency and constant linear accelerations need to be coupled to the appropriate rotational DOF. The lateral acceleration is coupled to the roll and the longitudinal acceleration is coupled to the pitch. There is no coupling between the vertical acceleration and the yaw. The low pass filter is designed such that the convolution of its transfer function and the associated high pass filter's transfer function is approximately unity. In this way, the frequency content of the resulting motion command for a linear acceleration is representative of the frequency content of the original signal. Rate limiting of the tilt coordination signal is used in order to avoid the perception of a miscue.

Another method to sustain linear accelerations is conceptualized in [36]. The proposed motion system has a continuous yaw capability. This would allow the simulator to be run in a centrifuge mode so that constant lateral, or in certain instances longitudinal, accelerations could be maintained indefinitely. The washout filter can be designed to implement this sort of mode instead of tilt coordination. The low passed filtered specific force can be used to determine certain parameters regarding the centrifuge mode such as the radius and rate of rotation.

Scaling and Limiting The washout will usually include some means of scaling the output of the vehicle dynamics. It is common to reduce the magnitude of the input to the washout by as much as half in

flight simulators. Attenuating the input by more than this will tend to reduce the effectiveness of the resulting motion, [64]. The effect of scaling is to effectively increase the performance envelope of the motion system by the inverse of the scale factor. The risk is that the subject may perceive that the accelerations are not of the magnitude that would be expected.

There are usually some sort of limit switches in hydraulic systems that will shut down the system if limits are exceeded. The washout often contains software limiting [57,58] that will avoid these limits so that panic shutdowns are avoided.

Two approaches to washout will be further investigated. The first is based on classical linear filter theory and is used to provide the basic frame work for conceptualizing a washout filter. The second is a filter based on an adaptive concept that overcomes some of the weaknesses of the classical linear filter.

Linear Filter Theory The development of a standard linear washout filter follows conventional digital filter design. In practice, a second or third order high pass filter is used.

The basic data flow for the linear filter is shown in Figure 4.5 following, [57,58,10,47,60]. The input to the filter is the specific force and the angular rate. The output is the high pass filtered specific force and a combination of the high pass filtered angular rate and *low pass* filtered specific force. The transfer function representation of a typical high pass filter for the translational DOF is, [57,58]:

$$F(s) = \frac{s^3}{(s^2 + 2 \zeta_{hp} \omega_{hp} s + \omega_{hp}^2)(s + \omega_{hp})} \quad (4.7)$$

Where ω_{hp} is the cutoff frequency and ζ_{hp} is the damping factor.

The complementary low pass filter is given by:

$$F(s) = \frac{\omega_{lp}^2}{s^2 + 2 \zeta_{lp} \omega_{lp} s + \omega_{lp}^2} \quad (4.8)$$

Where $\zeta_{lp} = \zeta_{hp}$ and $\omega_{lp} = 2 \omega_{hp}$

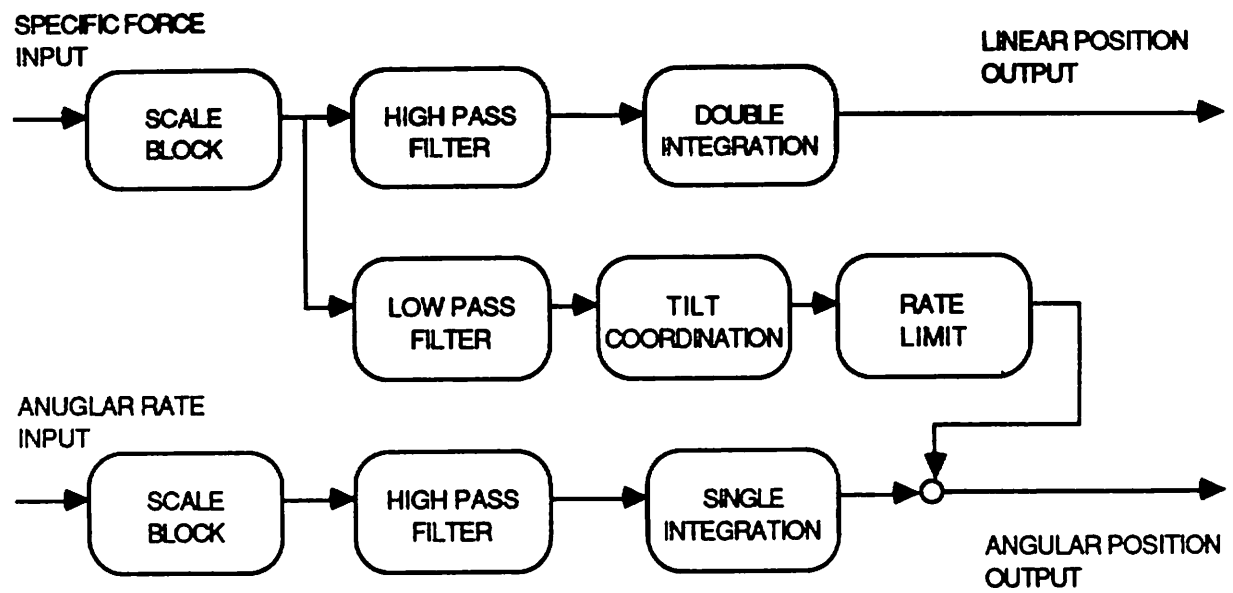


Figure 4.5: Block Diagram of a Linear Washout Filter

The resulting equations are commonly converted to the discrete domain by use of the bilinear z transform⁶ or by integrating the time domain equations numerically.

Adaptive Washout Filter The linear filter works well but can generate spurious false cues that can be very disconcerting to the subject in the simulator. A conceptual drawing showing the false cue is shown in Figure 4.6. Ariel and Sivan, [3], and Reid and Nahon, [57,58], present an adaptive filter structure that effectively reduces the false cue to imperceptible magnitudes. Parrish, [55,60] has a scheme as well but Ariel and Sivan show that it can distort the motion perception for some types of motion commands. The filter presented here follows the development by Reid and Nahon, [57,58].

The intent of an adaptive washout filter is to adapt the severity of the filter to the current state of the simulator. If there remains significant motion capability in the simulator, the filter need not restrict the motion commands at all. When limited capability remains, the filter needs to drastically attenuate all motion commands.

The design of an adaptive filter starts with defining the differential equation that relates input and output. The general form of the adaptive filter equation for translation is taken as [57,58,3]:

$$\frac{d^2 S}{dt^2} = p_1 a - k_1 S - k_2 \frac{dS}{dt} \quad (4.10)$$

Where:

- S = Displacement vector from the neutral position to the current simulator position
- a = Acceleration of the vehicle

⁶The bilinear z transform is:

$$s = \frac{2(1 - z^{-1})}{T(1 + z^{-1})} \quad (4.9)$$

Where T is the sample period. The advantage of the bilinear z transform is that it maps the left hand side of the s -plane completely into the unit circle of the z -plane. The disadvantage is that there is a frequency warping effect that must be taken into consideration, [53].

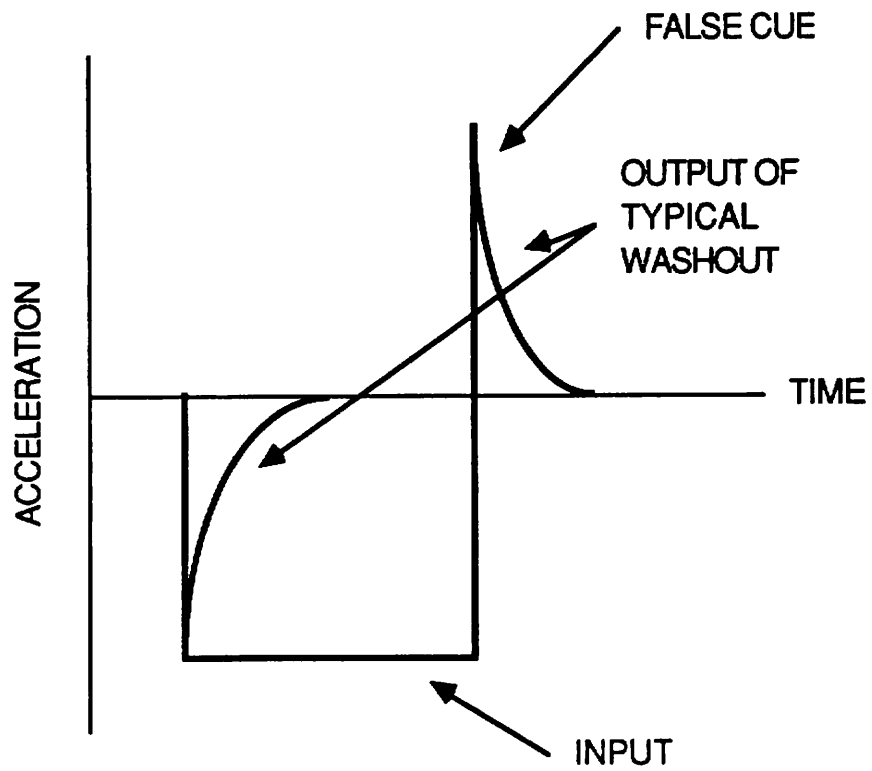


Figure 4.6: False cue generated with a step input to the high pass filter.

- p_1 = the adaptive parameter
- k_1, k_2 = constant weighting factors

For rotation [57,58,3]:

$$\frac{d\omega_s}{dt} = \ell(p_2 a) + p_3 \frac{d\omega_c}{dt} \quad (4.11)$$

Where:

- ω_s = angular velocity of the simulator
- ω_c = angular velocity of the vehicle
- p_2, p_3 = adaptive parameters
- a = acceleration of the vehicle
- $p_2 a$ = A term that represents tilt coordination.
- ℓ Refers to a non-linear limiting function that is applied to this particular term. It performs no limiting until the term reaches a pre-determined value then the limiting function clamps the value of this term at this limit. This clamping is maintained until the value falls below the limit.

A *cost function* is designed next. In this case, the cost function includes terms that define errors that should be minimized, and penalty terms that help increase the filtering effect in certain regions. The method of steepest descent is used to define the rate of change of the adaptive parameters. The cost function is differentiated with respect to the adaptive parameters and the resulting equations are used with the original differential equations to form a set of equations that are then solved numerically.

Differentiating the cost function has the effect of optimizing the filter equation for a given state of the input. The adaptive parameters are continually being updated so the filter is always tending towards an optimal design.

Here is a simplified example of the cost function for a translational DOF:

$$J_y = \frac{1}{2}[(a - \bar{S})^2 + (\dot{\phi}_{car} - \dot{\phi}_{sim})^2 + (\dot{S}^2 + S^2 + \dot{\phi}_{sim}^2 + \phi_{sim}^2) + (p_1 - p_{1o})^2 + (p_2 - p_{2o})^2 + (p_3 - p_{3o})^2] \quad (4.12)$$

Where:

- $a - \bar{S}$ = the error between the acceleration of the car and the simulator.
- $\dot{\phi}_{car} - \dot{\phi}_{sim}$ = the error between the roll velocity of the car and the simulator.
- $\dot{S}^2 + S^2 + \dot{\phi}_{sim}^2 + \phi_{sim}^2$ = A penalty weighting factor based on the linear velocity, linear position, angular velocity, and angular position of the simulator.
- $p_N - p_{No}$ = The difference between the initial value of the adaptive parameter and the current value.

4.1.4.2 Summary

In summary, the design of an effective simulator must weigh the required performance against the physical limitations of the technology at hand. Where the motion system lacks performance, proper use of washout can compensate to some degree. It's also possible to make use of other stimuli such as intense psychological loading, to compensate for limits in other systems.

The overall system must be matched to the expected maneuvers as well as the general vehicle performance envelope. Simply choosing the highest performance subsystems and connecting them is not enough. A balance must exist between the motion, visuals, and sound to produce the most ideal simulated environment.

4.1.5 Control Force Loading

An effective simulated environment does more than recreate visual, aural, and gross motion stimuli. There are other ways in which we interact with our environment when driving or flying. In particular,

we receive feedback through the controls that we use to maneuver a vehicle. The reaction forces and torques we are subjected to need to be recreated in the simulator environment in a realistic way.

There also has to be a link from the driver's controls to the input of the vehicle dynamics.

4.1.5.1 Vehicle Commands and Controls

The vehicle dynamics model needs to know what the driver is doing. Data acquisition is performed on the steering, brake, throttle, gear selection and any other controls that are available to maneuver the vehicle. Acquiring the command data is rather straight forward. Basic instrumentation is used and adapted to the needs of the particular model. These data are then made available to the vehicle dynamics at regular intervals.

The driver of a vehicle expects to have a certain reaction force on each of the controls. Sometimes this reaction is nothing more than a spring such as a throttle pedal. At other times, the force is linked to the dynamics of the vehicle such as for the steering. The correct forces need to be determined and then applied to the various controls.

Determining the control force for controls such as a throttle or clutch is straight forward. These systems are all spring loaded and adequate feedback is obtained from simply loading these controls with the appropriate springs and dampers.

The steering is a harder problem. The driver will always have a hand on the wheel and uses it to derive a lot of information about vehicle handling and road feel. This makes proper loading of the steering wheel a very important issue to proper and effective simulation.

The driver reacts forces and torques from the tires through the steering system. These forces and torques include an aligning torque from the wheel caster, lateral forces from cornering, and forces and torques from the tire-road interaction. This means that the vehicle dynamics needs to calculate these reactions as part of the regular cycle and communicate the resulting acceleration of the steering wheel to a control system for the steering.

In the particular dynamics formulation that we use, [22], the steering rack is modeled as one of the bodies. The acceleration, velocity

and displacement of the rack are then available for use at the end of each integration. The acceleration can be used as the setpoint for a servo control system connected to the steering wheel. The reaction to this acceleration command is then provided by the driver just as it would be in a real vehicle.

The transmission can be straight forward especially if it is an automatic transmission. In this case, no real feedback is present other than the shift lever position. This can be handled by the original shifting mechanism or by detente springs.

A manual transmission presents different problems. There is more feedback particularly if it is a floor mounted shift lever that enters the transmission directly. A fairly complicated control system is needed unless the real transmission can be used. Even then, the transmission must be driven at the input and output so that gear grinding and the proper feel when shifting is achieved. Fortunately, the transmission is used intermittently and the control force feedback is not critical to the simulation. This allows a lot of latitude in the transmission feedback system design and implementation.

The gages need to be driven as well. Parameters such as the speed, and rpm are available from the dynamics and drive train models. The oil pressure, water temperature, battery charging, and any other gages can normally be set to a nominal value and left alone.

4.1.6 The Total System

Integrating the simulator sub-systems properly presents a great challenge. Each of the separate parts interacts intimately with the whole. It is not enough to design each sub-system to perform to its maximum capability. Rather, it is better to design each component so that it functions with comparable effectiveness as all the rest. A bottom up design approach where pieces are put together without regard to overall system performance requirements does not work. The overall system performance must be considered and broken down into progressively smaller systems. This top down approach will insure that the final system will perform as wanted. The example of using a flight simulator motion base for a driving simulator is a good example. A flight simulator motion system has motion capabilities in keeping with

the motion of typical aircraft. The same motion base will be wholly inadequate for a land based vehicle. Its capabilities may exceed the needed motion in some DOF's for an automobile but will fail miserably for others.

An effective simulator system will be balanced mix of sensory stimulating devices. A system with modest capabilities in vision, motion, sound, and control loading will in general produce a better simulated environment than one that has a very expensive motion system at the expense of the CIG, sound and control loading.

4.1.7 The Future

The future of vehicle simulation is leading towards higher performance CIG's and vehicle dynamics. As hardware performance improves, it will open the door to CIG's that have higher scene detail, faster frame rates, and shorter pipeline delays. The vehicle dynamics will be able to take advantage of the faster throughput to include more detail in the modelling. For example, the addition of non-linear bushings, flexible bodies, and more complicated analytical tire models are but a few of the areas of current research.

The future holds great promise for the fidelity of real time simulation. New hardware, algorithms carefully designed to make use of parallel processing and new computing power, and more efficient mechanical systems will provide the means to achieve greater realism. The utility of simulators for training and engineering can do nothing but increase as performance goes up and cost goes down.

Chapter 5

References

5.1 References on Dynamics

Because the current vehicle models decomposes the car, or object or whatever into a set of linked rigid bodies, we must remember how the dynamics and kinematics of a single rigid body is analyzed. The best single reference on most topics in theoretical mechanics is Goldstein's book [30]. Beware that many misprints exist in the text and exercises.

An important reference for the foundations of the virtual work principle and recursive dynamics is Haug's excellent soon-to-be-published (?) text [37]. It is a very readable treatment on a less Olympian level than Goldstein, and even if less advanced (no treatment of Hamiltonians equations), it is more directly useful in understanding the tree-based recursive vehicle dynamics formulation. Haug, as well as his students, use the notation and terminology found in his text.

A good short treatment of dynamics¹ is Landau's book [45] of some hundred pages, each of which could be written in stone. Finally, among the mechanics texts, we must mention Lanczos' book [44] on variational principles in mechanics. It has has few formulas, but much insight, and it reads like a novel. In addition to these gems, there are many engineering and physics mechanics texts available to match any

¹Based almost exclusively on Lagrangian methods.

taste.

Because at times we've mentioned the underlying geometric and topological nature of mechanics, we've listed a few decent geometry references for the non-mathematician. To begin, an excellent text devoted to differential geometry for the scientist is the recent work by Burke [14]. The book is dedicated to "*all those who, like me, have wondered how in hell you can change $\dot{q}$ without changing q* " (an irresistible invitation). Reference [19] is a very complete treatment of modern methods in mathematical physics and includes an excellent section on differential geometry and manifolds. Another text on things differential geometric² is the famous older book by Synge and Schild [67]. We recommend avoiding the large tome by Abraham and Marsden [1], except possibly to look up specific topics once you are familiar with the heavy machinery surrounding all the physics.

²But from the classical "tensors as indexed objects" approach, thereby avoiding the more modern global methods.

5.2 References on Perception

The concept of washout is probably new to many readers. A good place to start is with Reid and Nahon, [57,58]. Thewe reports provides an excellent introductino to washout and human perception modelling. Riedel's, [60], work is similar but has differences in the exact formulation of the washout and perception models. Both references provide an excellent groundwork for understanding washout and related issues to simulator motion concerns.

Much work has been done in recent years regarding the physiology of the human perception systems. The two works, [12,13], edited by Boff are extremely detailed volumes on the physiology and operation of human perception. Regarding the actual modelling of vestibular systems, Riedel, [60], or Reid and Nahon, [57,58], are starting points for basic modelling issues. Greig, [32], has some new ideas, particularly regarding threshold levels, that are worth study by interested readers.

Bibliography

- [1] Abraham, R. and Marsden, J. *Foundations of Mechanics*, Revised Ed. Benjamin, 1978.
- [2] Alliant Computer Systems Corp. *Alliant FX/Series Product Summary*, June 1987.
- [3] Ariel, D. and Sivan, R. False cue reduction in moving flight simulators. In *IEEE Transactions on Systems, Man, and Cybernetics*, 1984.
- [4] Armstrong, W. Recursive Solution to the Equations of Motion of An N-Link Manipulator. In *Proc. 5th World Congress on Theory of Machines and Mechanism*, Vol. 2, (Montreal, 1979), 1343-1346.
- [5] Bae, D. and Haug, E. A Recursive Formulation for Constrained Mechanical System Dynamics. *Technical Report 86-14*, Center for Computer Aided Design, College of Engineering, University of Iowa, 1986.
- [6] Bae, D. and Haug, E. A Recursive Formulation for Constrained Mechanical System, Part I - Open Loop. *Mechanics of Structures and Machines*, 15 (1987), 359-382.
- [7] Bae, D. and Haug, E. A Recursive Formulation for Constrained Mechanical System, Part II - Closed Loop. *Mechanics of Structures and Machines*, 15 (1987).
- [8] Bae, D., Haug, E., and Kuhl, J. A Recursive Formulation for Constrained Mechanical System, Part III - Parallel Processor Implementation. *Mechanics of Structures and Machines*, 16 (1988).

- [9] Baumgarte, J. Stabilization of Constraints and Integrals of Motion in Dynamic Systems. *Computer Methods in Applied Mechanics in Engineering*, 1 (1972), 1-16.
- [10] Baarspul, M. *The Generation of Motion Cues on a Six-Degrees-of-Freedom Motion System*. Technical Report LR-248, Delft University of Technology, 1977.
- [11] Berthoz A., and Droulez, J. *Linear Self Motion Perception*. Technical Report, Laboratoire de Physiologie Neurosensorielle du CNRS, 1982.
- [12] Boff, K., Kaufman, L., and Thomas, J. (ed.). *Handbook of Perception and Human Performance*. Volume I and II, John Wiley and Sons, 1986.
- [13] Boff, K. and Lincoln, J. (ed.). *Engineering Data Compendium: Human Perception and Performance*. Volume I, II, and III, Harry G. Armstrong Aerospace Medical Research Laboratory, 1988.
- [14] Burke, W. *Applied Differential Geometry*. Cambridge University Press, 1985.
- [15] Campbell, C. *New Directions in Suspension Design*. Robert Bentley, Cambridge, Mass. 1981.
- [16] Casali, J. *Vehicular Simulation-Induced Sickness*. Volume I, II, and III, Naval Training Systems Center, 1986.
- [17] Casali, J. and Wierwille, W. The effects of various design alternatives on moving-base driving simulator discomfort. *Human Factors*, 22, 6 (1980), 741-756.
- [18] Chace, M. Methods and Experience in Computer Aided Design of Large-Displacement Mechanical Systems. In *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, E. Haug. (ed.). Springer-Verlag, Heidelberg, 1984.
- [19] Choquet-Bruhat, Y. and Dewitt-Morette, C. *Analysis, Manifolds, and Physics*, Revised Ed. North-Holland, 1982.

- [20] Clark, S. (ed.) *Mechanics of Pneumatic Tires*. U. S. Dept of Transportation, Washington, D. C., 1975.
- [21] Dennis, J. and More, J. Quasi-Newton Methods, Motivation and Theory. *SIAM Review* 19, 1 (1977), 46-89. September 1967, pp. 250-253.
- [22] Deyo, R., Briggs, J., and Doenges, P. Getting graphics in gear. *Computer Graphics*, 22, August 1988.
- [23] Dickmanns, E.D., and Zapp, A. Guiding Land Vehicles Along Roadways by Computer Vision. Presented at AFCET Conference "Automatique 85 - the tools for tomorrow", Toulouse, October 1985.
- [24] Doenges, Peter K. *Overview of Computer Image Generation in Visual Simulation*, Presented at ACM Siggraph Technical Courses, July, 1985.
- [25] Dorf, R. *Modern Control Systems*. Addison-Wesley Publishing Company, 1983.
- [26] Duncan, J.R., and Wegscheid, E.L. Off-road vehicle simulation for human factors research. *Paper 82-1610*, 1982 Winter Meeting, American Society of Agricultural Engineers (Chicago, Dec. 14-17, 1982.)
- [27] Drosdol, J. and Panik, F. The Daimler-Benz Driving Simulator: a Tool for Vehicle Development. *SAE Technical Paper Series*, 850334, (Feb. 1985).
- [28] Gear, W. Differential-Algebraic Equations. In *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, E. Haug (ed.) Springer-Verlag, Heidelberg, 1984.
- [29] Golub, G. and van Loan, C. *Matrix Computations*. John Hopkins Press, 1983.
- [30] Goldstein, H. *Classical Mechanics*. Addison-Wesley, Reading, Mass., 1980.

- [31] Greenwood, D. *Classical Dynamics*. Prentice-Hall, Englewood Cliffs, N.J., 1977.
- [32] Greig, G. *Masking Motion Cues by Random Motion: Comparison of Human Performance with a Signal Detection Model*. PhD thesis, University of Toronto, Aerospace Studies, 1988.
- [33] Hahn, S., and Kalb, E. The Daimler-Benz Driving Simulator Set-Up and Results of First Experiments. In *Summer Computer Simulation Conference Proceedings* Simulation Councils, Inc., San Diego, 1987, pp. 993 - 998.
- [34] Haug, E. *Computer Aided Analysis and Optimization of Mechanical System Dynamics*. Springer-Verlag, Heidelberg, 1984.
- [35] Haug, E. Elements and Methods of Computational Dynamics. In *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, E. Haug (ed.).
- [36] Haug, E. et. al. *Conceptual Design of a National Advanced Driving Simulator*. Technical Report, Center for Simulation and Design Optimization of Mechanical Systems, University of Iowa, 1989.
- [37] Haug, E. *Intermediate Dynamics*. Manuscript (to be published), 1986.
- [38] Hooker, W. and Margulies, G. The Dynamical Attitude Equations for an n - Body Satellite. *J. Astronaut. Sci.* 12 (1965), 123-128.
- [39] Issacs, P. and Cohen, M. Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics. *Computer Graphics* 21, 4 (July 1987), 215-224.
- [40] Kaeppler, W.D. Using a Driving Simulator to Investigate Vehicle Handling Properties. In *Human Decision Making and Manual Control* (H.P. Willumeit, Ed.), Elsevier Science Publishers B. V., North-Holland, 1986.

- [41] Kempf, D., Bonderson, L., and Slater, L. Real Time Simulation for Application to ABS Development. *SAE Technical Paper* 870336, 1987.
- [42] Kuo, B. *Automatic Control Systems*. Prentice-Hall, 1982.
- [43] Lambert, J. *Computational Methods in Ordinary Differential Equations*. John Wiley, New York, 1973.
- [44] Lanczos, C. *Variational Principles of Mechanics*. Dover, 1986.
- [45] Landau, L. and Lifshitz, E. *Mechanics*, Third Ed. Pergamon Press, 1976.
- [46] Luh, J., Walker, M., and Paul, R. On-line Computational Scheme for Mechanical Manipulator. *J. Dyn. Syst. Measurement Control* 102 (1980), 69-76.
- [47] Liu, Z. *A Study of Washout Filters for a Simulator Motion Base*. Technical Report 246, University of Toronto, Aerospace Studies, 1983.
- [48] Mayer, N. and M. Cosman. Enhancing the Computer Generated Illusion. *Proceedings of the Interservice Industry Training Equipment Conference*, November, 1982.
- [49] McHenry, R. Automobile Dynamics - A computer Simulation of General Three-Dimensional Motions. *SAE Technical Paper* 710361, 1971.
- [50] Mitsche, M. *Dynamik der Kraftfahrzeuge*. Springer-Verlag, Berlin, 1982.
- [51] Nikravesh, P. and Chung, I. Applications of Euler Parameters to the Dynamic Analysis of Three-Dimensional Constrained Mechanical Systems. *J. Mechanical Design* 104 (1982), 785-791.
- [52] Nordmark, S. VTI Driving Simulator - Mathematical Model of a Four-wheeled Vehicle for Simulation in real Time. *VTI Rapport* Nr 267A 1984. Statens vag-och trafikinstitut, Linloping, Sweden, 1984.

- [53] Oppenheim, A., and Schafer, R. *Digital Signal Processing*. Prentice Hall, 1975.
- [54] Orlandea, N., Chace, M., and Calahan, D. A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems- Parts 1 and 2. *J. of Engineering for Industry* (1977), 773-784.
- [55] Parrish, R., Dieudonne, J., Bowels, R., and Martin, Jr., D. *Coordinated Adaptive Washout for Motion Simulation*. Technical Report 73-930, AIAA, 1973.
- [56] Regan, D., Kaufman, L., and Lincoln, J. Motion in depth and visual acceleration. In *Engineering Data Compendium: Human Perception and Performance*, chapter 19, Harry G. Armstrong Aerospace Medical Research Laboratory, 1986.
- [57] Reid, L., and Nahon, M. *Flight Simulation Motion-Base Drive Algorithms: Part 1 - Developing and Testing the Equations*. Technical Report 296, University of Toronto, Aerospace Studies, 1986.
- [58] Reid, L., and Nahon, M. *Flight Simulation Motion-Base Drive Algorithms: Part 2 - Selecting the System Parameters*. Technical Report 307, University of Toronto, Aerospace Studies, 1986.
- [59] Richter, Bernd. Driving simulator studies — the influence of vehicle parameters on safety in critical situations. SAE Technical Paper Series, 741105, (Feb. 1974).
- [60] Riedel, S. *A Comparison of Washout Filters using a Human Dynamic Orientation Model*. Master's thesis, Massachusetts Institute of Technology, 1977.
- [61] Roberson, R. and Schwertassek, R. *Dynamics of Multibody Systems*. Springer-Verlag, New York, 1988.
- [62] Schumacker, R. A. *A New Visual System Architecture*. Proceedings from the 2nd Interservice Industry Training Equipment Conference, November, 1980.

- [63] Sheridan, T.B., Paynter, H.M., and Coons, S.A. *Some novel Display Techniques for Driving Simulation*. In IEEE Transactions on Human Factors in Electronics, Vol. HFE-5, September 1964, pp. 29-32.
- [64] Sinacori, J. *The Determination of Some Requirements for a Helicopter Flight Research Simulation Facility*. Technical Report CR-152066, NASA, 1977.
- [65] Stapleford, R.L., and Klein, R.H. Recent Developments in Driving Simulator Hardware and Software. *Paper No. 273, Systems Technology, Inc., Hawthorne, CA*. Prepared for the TRB Committee Workshop on the Use and Abuse of Driving Simulators, 24th Annual Meeting, Human Factors Society, Los Angeles, Oct. 13-17, 1980.
- [66] Strang, G. *Linear Algebra and its Applications*, Second Ed. Academic Press, 1980.
- [67] Synge, J. and Schild, A. *Tensor Calculus*. Dover Publications, 1978.
- [68] Wehage, R. and Haug, E. Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Mechanical Systems. *J. of Mechanical Design*, 104 (1982), 247-255.
- [69] Weirwille, W.W., Gagne, G.A., and Knight, J. R. A Laboratory Display System Suitable for Man-Machine Interface. In *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 3.
- [70] Wittenburg, J. *Dynamics of Systems of Rigid Bodies*. B. G. Teubner, Stuttgart, 1977.